PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-353678
(43)Date of publication of application : 25.12.2001

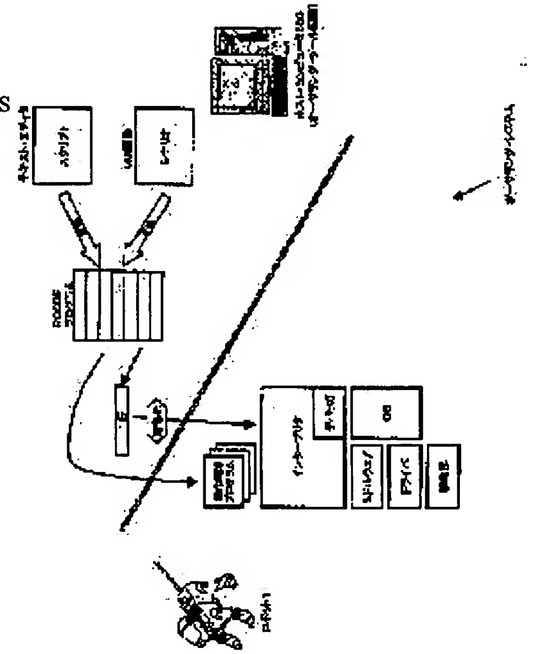
(51)Int.Cl. B25J 13/00
B25J 5/00
B25J 9/16
B25J 9/22
B25J 19/00

(21)Application number : 2000-175159
(22)Date of filing : 12.06.2000

(71)Applicant : SONY CORP
(72)Inventor : KASUGA TOMOAKI
OKITA FUMIKO

(54) AUTHORING SYSTEM AND METHOD AND STORAGE MEDIUM

(57)Abstract:
PROBLEM TO BE SOLVED: To originate an operating pattern by use of an assembly of parts which specify the operating condition of a robot and a process routine.
SOLUTION: A user originates and edits a specified scenario of a robot 1 through a mouse operation using a GUI screen. An authoring tool converts the scenario originated and edited into a mnemonic mode called RCODE. During debugging of an RCODE operation control program, an RCODE program is taken out row by row and enciphered for transfer to the robot by use of a wireless communications means. At the robot, an interpreter interprets and executes the RCODE and performs debugging.



LEGAL STATUS
[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

362-100 PCT/US
English Translation: JP2001-353678

* NOTICES *
JPO and INPIT are not responsible for any damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The authoring system which is an authoring system for creating and editing a robot's control program of operation, and is characterized by providing the editorial department which offers the workplace for creating and editing this robot's scenario of operation to a user, and the transducer which changes creation and the edited scenario into the program code which can be interpreted on said robot through said editorial department.
[Claim 2] Said editorial department is an authoring system according to claim 1 characterized by providing a user with the components group which components-izes the operating state used frequently, a manipulation routine, or its chicken type, and prepares it, and the actuation screen which selects each part article of said components group by directions actuation through a coordinate designating device, and is arranged.
[Claim 3] Said editorial department is an authoring system according to claim 1 characterized by providing a user with the environment where the script of a predetermined programming language format is created and edited.
[Claim 4] Said transducer is an authoring system according to claim 1 characterized by changing the scenario of operation described in the scenario of operation and/or script format which were expressed by arrangement of components into the mnemonic code which can be interpreted in said robot.
[Claim 5] Furthermore, the authoring system according to claim 1 characterized by having the means of communications for transmitting the program code changed in said transducer to said robot.
[Claim 6] Said means of communications is an authoring system according to claim 5 characterized by performing said robot and wireless data transmission.
[Claim 7] Said means of communications is an authoring system according to claim 5 characterized by taking out at a time one step of program codes changed in said transducer, and transmitting to said robot.
[Claim 8] Said means of communications is an authoring system according to claim 5 characterized by enciphering the program code changed in said transducer, and transmitting to said robot.
[Claim 9] Furthermore, the authoring system according to claim 5 characterized by having the interpretation section which interprets the program code received through said means of communications, and the drive control section which drives said robot

according to the interpretation result by said interpretation section on said robot.

[Claim 10] Said interpretation section is an authoring system according to claim 9 characterized by interpreting the received program code per 1 step, and performing it.

[Claim 11] said drive control section be the authoring system according to claim 9 characterize by what change into the format in which said robot have the posture transition limit information about the posture whose **** transition be attain via the actuation and the posture in the actuation which can be transit directly from a current posture , or a posture list , and posture transition [contents / of directions / of a program code] be possible based on this posture transition limit information .

[Claim 12] it be the authoring system according to claim 11 which said posture transition limit information be hold in the directed graph format which consist of arcs of operation which connect between the node which show the posture which said robot can take , and two nodes which can be change , and be characterize by what said drive control section search for this directed graph , and the contents of directions of a program code be change into the format in which posture transition be possible for .

[Claim 13] The authoring approach characterized by to provide the edit step which is the authoring approach for creating and editing a robot's control program of operation, and offers the workplace for creating and editing this robot's scenario of operation to a user, and the conversion step which change creation and the edited scenario into the program code which can be interpreted on said robot through said edit step.

[Claim 14] The authoring approach according to claim 13 characterized by providing a user with the components group which components-izes the operating state used frequently, a manipulation routine, or its chicken type, and prepares it at said edit step, and the actuation screen which selects each part article of said components group by directions actuation through a coordinate designating device, and is arranged.

[Claim 15] The authoring approach according to claim 13 characterized by providing a user with the environment where the script of a predetermined programming language format is created and edited, at said edit step.

[Claim 16] The authoring approach according to claim 13 characterized by changing the scenario of operation described at said conversion step in the scenario of operation and/or script format which were expressed by arrangement of components into the mnemonic code which can be interpreted in said robot.

[Claim 17] Furthermore, the authoring approach according to claim 13 characterized by having a communication link step for transmitting the program code changed in said conversion step to said robot.

[Claim 18] The authoring approach according to claim 17 characterized by performing said robot and wireless data transmission at said communication link step.

[Claim 19] Said communication link step is the authoring approach according to claim 17 characterized by taking out at a time one step of program codes changed in said conversion step, and transmitting to said robot.

[Claim 20] The authoring approach according to claim 17 characterized by enciphering the program code changed in said transducer at said communication link step, and transmitting to said robot.

[Claim 21] Furthermore, the authoring approach according to claim 17 characterized by performing the interpretation step which interprets the program code received through said communication link step, and the drive control step which drives said robot according to the interpretation result by said interpretation step on said robot.

[Claim 22] The authoring approach according to claim 21 characterized by interpreting the received program code per 1 step, and performing it at said interpretation step.

[Claim 23] said robot be the authoring approach according to claim 21 characterize by what it have the posture transition limit information about the posture whose **** transition be attain via the actuation and the posture in the actuation which can be transit directly from a current posture , or a posture list , and the contents of a program code of directions change into the format in which posture transition be possible for based on this posture transition limit information at said drive control step .

[Claim 24] said posture transition limit information be an authoring approach according to claim 23 characterize by what it be hold in the directed graph format which consist of arcs of operation which connect between the node which show the posture which said robot can take , and two nodes change , and said drive control step be search for this directed graph , and the contents of a program code of directions be change into the format in which posture transition be possible for .

[Claim 25] It is the storage which stored physically the computer software described to perform authoring processing for created and editing a robot's control program of operation on computer system in the computer-readable format. The edit step which said computer software provides with the workplace for creating and editing this robot's scenario of operation to a user, The storage characterized by providing the conversion step which changes creation and the edited scenario into the program code which can be interpreted on said robot through said edit step.

[Claim 26] The storage according to claim 25 characterized by providing a user with the components group which components-izes the operating state used frequently, a manipulation routine, or its chicken type, and prepares it at said edit step, and the actuation screen which selects each part article of said components group by directions actuation through a coordinate designating device, and is arranged.

[Claim 27] The storage according to claim 25 characterized by providing a user with the environment where the script of a predetermined programming language format is created and edited, at said edit step.

[Claim 28] The storage according to claim 25 characterized by changing the scenario of operation described at said conversion step in the scenario of operation and/or script format which were expressed by arrangement of components into the mnemonic code which can be interpreted in said robot.

[Claim 29] Furthermore, the storage according to claim 25 characterized by having a communication link step for transmitting the program code changed in said conversion step to said robot.

[Translation done.]

2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the authoring system and approach for creating the data according to a predetermined scenario, and relates to the authoring system and approach of creating a series of command/data which describe a robot's predetermined pattern of operation especially.

[0002] Furthermore, in detail, this invention relates to the authoring system and approach of creating a pattern of operation using the set of the components which specify a robot's operating state, and relates to the authoring system and approach of arranging each part article on a computer display, and creating a pattern of operation especially.

[0003]

[Description of the Prior Art] The thing of the machinery which performs movement modeled on actuation of human being using the electric or magnetic operation is called "robot." It is said that a robot's origin of a word originates in ROBOTA (slave machine) of a slab word. Although it was in our country that a robot began to spread from the end of the 1960s, the many were the industrial robots (industrial robot) in works aiming at automation, full automation, etc. of production, such as a manipulator and a carrier robot.

[0004] Recently, the researches and developments about leg formula mobile robots, such as a robot (humanoid robot) of "the human form" which imitated the body mechanism of the animal which performs 2-pair-of-shoes walks in erect posture, such as a pet mold robot which imitated the body mechanism of the animal of quadrapedalism and its actuation like a dog or a cat or Homo sapiens, and an ape, and actuation, or "a human mold", or the stable walk control of those progress, and the expectation for utilization has also been growing. although these leg formula mobile robot compares with a crawler type robot, it is unstable and attitude control and walk control become difficult -- rise and fall of a stairway and an obstruction -- getting over -- etc. -- it excels in the point that flexible walk / transit actuation is realizable.

[0005] Like an arm type robot, the robot of a deferment type which is implanted and used for a certain specific location works only in fixed and local workspaces, such as assembly, a sorting activity, etc. of components. On the other hand, the robot of workspace of a portable type is un-restrictive, and he can move free in a predetermined path or non-path top, and the human activity of predetermined or arbitration can be executed by proxy, or he can offer the various services which replace Homo sapiens, a dog, or other life objects.

[0006] As one of the applications of a leg formula mobile robot, vicarious execution of various kinds of difficulty activities in an industrial activity, a production activity, etc. is mentioned. For example, it is vicarious execution of the maintenance in a nuclear power plant, a thermal power station plant, and a petrochemical plant, conveyance and assembly operation of the components in a plant, cleaning in a skyscraper, and the risk activity and difficulty activity like the rescue in a fire site and others etc.

[0007] Moreover, the application of "symbiosis" or "entertainment" of a life adhesion mold, i.e., human being, is mentioned rather than above-mentioned activity exchange as other applications of a leg formula mobile robot. This kind of robot emulates the rich feeling expression using the mechanisms of operation and the limbs of a leg formula ambulatory exercise with comparatively high intelligence, such as Homo sapiens or a dog (pet). Moreover, it is required it not only performs faithfully the pattern of operation inputted beforehand, but that it should realize the lively response expression in response to a partner's language and attitudes ("it strikes"). ["it praises" or "he scolding",]

[0008] The conventional toy machine has the fixed relation between user actuation and response actuation, and cannot change actuation of a toy according to liking of a user. Consequently, a user becomes ***************** soon about the toy which repeats only the same actuation.

[0009] On the other hand, the intellectual robot has the model resulting from actuation, and realizes autonomous thinking and motion control by changing a model based on input, such as voice from the outside, and an image, a tactile sense, and opting for actuation. When a robot prepares a feeling model and an instinct model, the autonomous action according to a robot's own feeling and instinct can be expressed. Moreover, when a robot equips a picture input device and voice-input/output equipment and performs image recognition processing and speech recognition processing, it also becomes possible to realize realistic communication with human being on more advanced intellectual level.

[0010] Moreover, by answering having detected the stimulus from the outside, such as user actuation, and changing this model, namely, giving the "study effectiveness", or it does not get bored for a user, the pattern of operation which was adapted for liking can be offered.

[0011] The leg formula mobile robot of these days has high information processing capacity, and can regard as a kind of computing system. Therefore, the altitude and a series of complicated operating sequences which are constituted by the combination of the pattern of operation realized on a robot or two or more fundamental patterns of operation are built according to the same activity as computer programming.

[0012] Moreover, a robot's diffusion rate will increase increasingly from now on, and permeating deeply not only the industrial world but ordinary homes and everyday life will be expected. About the product which pursues entertainment nature, it is especially expected [that a consuming public layer without the advanced knowledge about a computer or computer programming purchases and uses a robot in many cases, and]. It is thought desirable to offer the tool which supports the thing for creating and editing a robot's operating sequence comparatively easily and efficiently by interactive processing also for such a general user, i.e., an authoring system.

[0013]

[Problem(s) to be Solved by the Invention] The purpose of this invention is to offer the outstanding authoring system and outstanding approach of creating a series of command/data which describe a robot's predetermined pattern of operation.

[0014] The further purpose of this invention is to offer the outstanding authoring system and outstanding approach of creating a pattern of operation using the set of the components which specify a robot's operating state.

[0015] The further purpose of this invention is to offer the outstanding authoring system and outstanding approach of arranging each part article on a computer display, and creating a pattern of operation.

[0016]
[Means for Solving the Problem] This invention is made in consideration of the above-mentioned technical problem. The 1st side face The editorial department which is the authoring system or approach for creating and editing a robot's control program of operation, and offers the workplace for creating and editing this robot's scenario of operation to a user, or an edit step, It is the authoring system or approach characterized by providing the transducer which changes creation and the edited scenario into the program code which can be interpreted on said robot through said editorial department, or a conversion step.

[0017] You may make it said editorial department or an edit step provide a user with the components group which components-izes the operating state used frequently, a manipulation routine, or its chicken type, and prepares it, and the actuation screen which selects each part article of said components group by directions actuation through a coordinate designating device, and is arranged.

[0018] Or you may make it said editorial department or an edit step provide a user with the environment where the script of a predetermined programming language format is created and edited.

[0019] Moreover, you may make it said transducer or transducer change the scenario of operation described in the scenario of operation and/or script format which were expressed by arrangement of components into the mnemonic code which can be interpreted in said robot. An example of the mnemonic code said here is RCODE. The after-mentioned is yielded about the detail of RCODE.

[0020] Moreover, you may have further the means of communications or the communication link step for transmitting the program code changed in said transducer or the conversion step to said robot. At means of communications or a communication link step, short-distance wireless data transmission, such as bluetooth and .11B, is applicable, for example.

[0021] The means of communications or the communication link step said here takes out at a time one step of program codes changed in said transducer or the conversion step, and you may make it transmit them to said robot. On the other hand, in a robot side, the program code received using the interpreter can be interpreted serially, and can be performed and/or debugged.

[0022] Moreover, in order to maintain the security of commo data, said means of communications or a communication link step enciphers the program code changed in said transducer or the conversion step, and you may make it transmit it to said robot.

[0023] Moreover, you may have the interpretation section which interprets the program code received through said means of communications or a communication link step or an interpretation step, and the drive control section which drives said robot according to the interpretation result by said interpretation section or the interpretation step or a drive control step on said robot.

[0024] In such a case, the received program code is interpreted per 1 step, and said interpretation section or an interpretation step may be made to perform it.

[0025] Generally a leg formula robot has the actuation and the posture which can be transited directly from a current posture, and the posture whose **** transition is attained via a certain actuation and posture. The robot concerning this invention may have beforehand the posture transition limit information about the posture whose **** transition is attained via the actuation and the posture in the actuation which can be transited directly from a current posture, or a posture list.

[0026] In such a case, said drive control section or a drive control step can change the contents of directions of a program code into the format in which posture transition is possible based on this posture transition limit information.

[0027] Moreover, this posture transition limit information can be held in the directed graph format which consists of arcs of operation which connect between the node which shows the posture which said robot can take, and two nodes which can be changed. Therefore, said drive control section or a drive control step can change the contents of directions of a program code into the format in which posture transition is possible easily by searching for this directed graph.

[0028] Moreover, the 2nd side face of this invention is the storage which stored physically the computer software described to perform authoring processing for created and editing a robot's control program of operation on computer system in the computer-readable format. The edit step which said computer software provides with the workplace for creating and editing this robot's scenario of operation to a user, It is the storage characterized by providing the conversion step which changes creation and the edited scenario into the program code which can be interpreted on said robot through said edit step.

[0029] The storage concerning the 2nd side face of this invention is a medium which offers computer software physically in a computer-readable format to the general purpose computer system which can perform various program codes, for example. Attachment and detachment of CD (Compact Disc), FD (Floppy Disc), MO (Magneto-Opticaldisc), etc., etc. are free for such a medium, and it is a storage of portability. Or it is also technically possible to provide specific computer system with computer software in a computer-readable format via transmission media, such as a network (for a network not to ask distinction of wireless and a cable), etc.

[0030] Such a storage defines the collaboration-relation on the structure of the computer software and the storage for realizing the function of computer software predetermined in a computer system top, or a function. If it puts in another way, by installing predetermined computer software in computer system through the storage concerning the 2nd side face of this invention, on computer system, a collaboration-operation is demonstrated and the same operation effectiveness as the authoring system or approach concerning the 1st side face of this invention can be acquired.

[0031]

[Function] A user uses the GUI screen which the authoring tool which realized this invention offers, and creates and edits the scenario of a convention of a robot by mouse actuation. Or a user may use a text editor etc., and may create and edit a robot's control program of operation in a script format.

[0032] An authoring tool changes the scenario and the control program of a script format of operation which were created and edited into the mnemonic code called RCODE. At the time of debugging of a RCODE actuation control program, a RCODE program is taken out for every line, and it enciphers, and transmits to a robot side serially using radio means, such as bluetooth and .11B.

[0033] In the robot side of another side, it has an interpreter/debugger, middleware, a driver, etc. as activation and the debugging environment of the control program of

operation described by RCODE etc.
[0034] An interpreter is a high-level-language program which reads at a time the program of one line described in the RCODE format, interprets it, and performs it. A debugger discovers the error in a RCODE program (bug), and is a program which supports the activity which corrects. Middleware is a processing engine which performs control of leg formula actuation of the walk and others in a robot etc., recognition processing of the input image from a camera, recognition processing of the voice input from a microphone, etc.
[0035] Middleware has registered beforehand the posture in which a robot 1 can change, and the actuation at the time of changing in the form of a directed graph, changes the action command from a high order program into posture transition information according to a directed graph, and publishes the concrete directions of operation to a driver.
[0036] Therefore, a general user does not need to care about detailed hardware information like the posture and the transition property of operation in an actual robot, as long as it edits a robot's 1 scenario of operation or a control program of operation is described in a script format on a GUI screen using an authoring tool. Moreover, since the difference between logical posture and directions of operation on RCODE command level, and the posture and the transition property of operation on hardware is absorbed in middleware level, in it being contrary to a robot's posture and transition property of operation, the object of debugging of a RCODE program does not become.
[0037] The purpose, the description, and advantage of further others of this invention will become [rather than] clear by detailed explanation based on the example and the drawing to attach of this invention mentioned later.
[0038]
[Embodiment of the Invention] Hereafter, the example of this invention is explained in detail, referring to a drawing.
[0039] A. The appearance configuration of the mobile robot 1 which performs the leg formula walk by the limbs with which operation is presented in this invention is shown in the block diagram 1 of an authoring system. This robot 1 is a mobile robot of the multi-joint mold constituted by using as a model the configuration and structure of an animal of having the limbs as illustration. User actuation was answered and the mobile robot 1 of this example can especially do an expression of operation while he has a side face of the pet mold robot designed by imitating the configuration and structure of a dog which are the example of representation of a pet, for example, coexists with human being in human being's living conditions.
[0040] A mobile robot 1 consists of, the idiosoma unit 2, a head unit 3, a tail 4, and Limbs 6A-6D, i.e., leg units.
[0041] The head unit 3 is arranged in the upper limit before abbreviation of the idiosoma unit 2 through the neck joint 7 with a roll, a pitch, and the degree of freedom of each shaft orientations (illustration) of a yaw. Moreover, the CCD (Charge Coupled Device: charge-coupled device) camera 15 equivalent to the "eye" of a dog, the microphone 16 equivalent to a "lug", the loudspeaker 17 equivalent to "opening", and the touch sensor 18 equivalent to tactile feeling are carried in the head unit 3. The sensor which constitutes a living body's senses besides these may be included.
[0042] The tail 4 is attached in the abbreviation Gokami edge of the idiosoma unit 2 free [a curve or rocking] through the tail joint 8 with the degree of freedom of a roll and a

pitching axis.

[0043] The leg units 6A and 6B constitute a forefoot, and the leg units 6C and 6D constitute hind legs. Each leg units 6A-6D consist of combination of the femoral region units 9A-9D and the leg part units 10A-10D, and are attached in each corner of front and rear, right and left of idiosoma unit 2 base, respectively. The femoral region units 9A-9D are connected with each predetermined part of the idiosoma unit 2 by the hip joints 11A-11D with a roll, a pitch, and the degree of freedom of each shaft of a yaw. Moreover, it is connected by the knee joints 12A-12D with the degree of freedom of a roll and a pitching axis between the femoral region units 9A-9D and the leg part units 10A-10D.

[0044] the mobile robot 1 constituted like illustration making the head unit 3 shake vertically and horizontally by driving each joint actuator by the command from a control section mentioned later, or making a tail 4 wag **** -- each -- a foot -- Units 6A-6D are made to drive cooperatively, and actuation of a walk, transit, etc. can be realized.

[0045] In addition, in fact, a mobile robot's 1 joint degree of freedom is arranged for every shaft, and is offered by the rotation drive of a joint actuator (not shown). Moreover, the number of the joint degree of freedom which a mobile robot 1 has is arbitrary, and does not limit the summary of this invention.

[0046] The block diagram of the electrical and electric equipment and control network of a mobile robot 1 is typically shown in drawing 2 . As shown in this drawing, a mobile robot 1 consists of the control section 20 which performs generalization-control of the whole actuation, and other data processing, the I/O section 40, a mechanical component 50, and a power supply section 60. Hereafter, each part is explained.

[0047] The I/O section 40 contains various kinds of sensors equivalent to the senses, such as CCD camera 15 which is equivalent to a mobile robot's 1 eyes as the input section, and the microphone 16 equivalent to a lug, the touch sensor 18 equivalent to tactile feeling. Moreover, the loudspeaker 17 equivalent to opening etc. is equipped as the output section. These output section can express the user feedback from a robot 1 in the form of [other than a machine movement pattern with a foot etc.].

[0048] A mobile robot 1 can recognize the objective configuration and the color of arbitration which exist on workspace by a camera 15 being included. Moreover, the mobile robot 1 may have further the receiving set which receives dispatch waves other than a vision means like a camera, such as infrared radiation, an acoustic wave, a supersonic wave, and an electric wave. In this case, based on the sensor output which detects each carrier wave, the location and sense from the source of dispatch are measurable.

[0049] A mechanical component 50 is functional block which realizes machine movement of a mobile robot 1 according to the predetermined movement pattern which a control section 20 orders it, and consists of drive units prepared for every shafts, such as a roll in each joint, such as the neck joint 7, the tail joint 8, hip joints 11A-11D, and knee joints 12A-12D, a pitch, and a yaw. A mobile robot 1 has the joint degree of freedom of n pieces, therefore a mechanical component 50 is constituted from the example of illustration by n drive units. Each drive unit consists of combination of the driver 53 which controls the rotation location and rotational speed of a motor 51 accommodative based on the output of the motor 51 which performs rotation actuation of the circumference of a predetermined shaft, the encoder 52 which detects the rotation location of a motor 51, and an encoder 52.

[0050] A power supply section 60 is a functional module which supplies electric power to each electrical circuit in literal [the] and a mobile robot 1 etc. The mobile robot 1 concerning this example is the autonomous drive type which used the dc-battery, and a power supply section 60 consists of a charge dc-battery 61 and a charge-and-discharge control section 62 which manages the charge-and-discharge condition of the charge dc-battery 61.

[0051] The charge dc-battery 61 consists of gestalten of the "battery pack" which package-ized two or more nickel cadmium battery cels to the cartridge-type.

[0052] Moreover, the charge-and-discharge control section 62 grasps the remaining capacity of a dc-battery 61 by measuring the terminal voltage of a dc-battery 61, charge/strength of discharge current, the ambient temperature of a dc-battery 61, etc., and determines an initiation stage, a termination stage, etc. of charge. The initiation and the termination stage of charge which the charge-and-discharge control section 62 determines are notified to a control section 20, and serve as a trigger for a mobile robot 1 to start and end charge operation. The after-mentioned is yielded about the detail of charge operation.

[0053] A control section 20 is equivalent to "brains", for example, is carried in a mobile robot's 1 head unit 3 or idiosoma unit 2.

[0054] The configuration of a control section 20 is further illustrated in the detail at drawing 3 . As shown in this drawing, the control section 20 has the composition that the bus connection of CPU (Central Processing Unit)21 as a Maine controller was carried out to each circuit component, which are memory and others, or a peripheral device. The address (a memory address or I/O Address) of a proper is assigned at each to each equipment on a bus 27, and CPU21 can communicate with the specific equipment on a bus 28 by addressing.

[0055] RAM (Random Access Memory)22 is the memory which consisted of volatile memory, such as DRAM (Dynamic RAM), and which can be written in, loads the program code which CPU21 performs, or is used for temporary preservation of activity data.

[0056] ROM (Read Only Memory)23 is a read-only memory which stores a program and data everlastingly. The self-test test program performed to a mobile robot's 1 power up, the control program which specifies actuation of a mobile robot 1 are mentioned to the program code stored in ROM23.

[0057] The "sensor input-process program" which processes sensor inputs, such as a camera 15 and a microphone 16, the "action instruction program" which generates a mobile robot's 1 action, i.e., a movement pattern, based on a sensor input and a predetermined model of operation, and the "drive control program" etc. which controls the drive of each motor, the voice output of a loudspeaker 17, etc. according to the generated movement pattern are contained in a robot's 1 control program. High actuation of entertainment nature, such as a "hand", a "rain check", "stability", and utterance of the cry of animals, such as "one one", may be included in the movement pattern generated in addition to usual locomotion and usual transit movement.

[0058] Moreover, creation and various kinds of edited operating-sequence programs are included, using an authoring tool as a control program of others of a robot 1. Although an authoring tool is started under a software execution environment predetermined in an external computer system top, it is explained in detail to the program created and edited by the authoring tool list on this tool, therefore the back.

[0059] Like EEPROM (Electrically Erasable and Programmable ROM), nonvolatile memory 24 consists of memory devices in which elimination re-writing is possible electrically, and it is used in order to hold the data which should be updated serially in un-volatilizing. The model which specifies a mobile robot's 1 behavior pattern is mentioned to the data which should be updated serially.

[0060] An interface 25 is equipment for interconnecting with the device besides a control section 20, and making the data exchange possible. An interface 25 performs data I/O between a camera 15, a microphone 16, and a loudspeaker 17. Moreover, an interface 25 is each driver 53-1 in a mechanical component 50. -- I/O of data or a command is performed in between. Moreover, an interface 25 can also perform transfer of charge initiation and a charge terminate signal among power supply sections 60.

[0061] An interface 25 Serial interface, such as RS(Recommended Standard)-232C, Parallel interfaces, such as IEEE (Institute of Electrical and electronics Engineers)1284, A USB (Universal Serial Bus) interface, An i-Link (IEEE1394) interface, a SCSI (Small Computer System Interface) interface, etc. It has a general interface for peripheral-device connection of a computer, and may be made to perform program and migration of data between the external instruments by which local connection was made.

[0062] Moreover, it has an infrared-ray-communication (IrDA) interface as one of the interfaces 25, and may be made to perform an external instrument and radio. As for the transceiver section for infrared ray communication, it is desirable from a viewpoint of receiving sensibility to be installed in the point of mobile-robot 1 bodies, such as the head unit 2 and a tail 3.

[0063] Furthermore, a control section 20 can perform an external host computer 100 and data communication via contiguity radio as shown in "bluetooth" or ".11B" or LAN (Local Area Network: (trademark), for example, Ethernet), or the Internet including radio interface 26 Network Interface Card (NIC) 27.

[0064] The purpose of the data communication between such a robot and a host computer is operating actuation of a mobile robot 1 by remote control using a remote computer resource. Moreover, other purposes of this data communication are to supply data and program of the robots 1, such as a model of operation and other program codes, required for motion control to a mobile robot 1 via a network. Moreover, other purposes of this data communication are debugging processings of the real time according using an authoring tool to collaboration-actuation with download of the program for creation and the robot motion control which edited, the host computer 100 of such a program for motion control, and a robot 1 on a host computer 100. However, about an authoring tool or debugging processing, it mentions later behind.

[0065] Moreover, a control section 20 may be equipped with the keyboard 29 which consists of a ten key and/or an alphabet key. A keyboard 29 is used for the direct command input in a work site, and also it is used for the input of owner authentication information, such as a password.

[0066] The robot 1 concerning this example can perform autonomous actuation, when a control section 20 performs a predetermined control program. Moreover, while having an input unit equivalent to the senses of human beings, such as an image input (namely, camera 15), voice input (namely, microphone 16), and a touch sensor 18, or an animal, it has the intelligence which performs reasonable or emotional actuation which answered these external inputs.

[0067] The mobile robot 1 constituted as shown in drawing 1 - drawing 3 has the following descriptions. Namely, [0068] (1) When changing from a certain posture to other postures is directed, don't change between each posture directly but change via an in-between posture without the unreasonableness prepared beforehand.
(2) A notice can be received when the posture of arbitration is reached by posture transition.
(3) In each unit unit, such as a head, a foot, and the tail section, a posture can be managed independently and can carry out attitude control. Namely, apart from the whole robot's 1 posture, a posture is manageable for every unit.
(4) A parameter to show the detail of actuation of an instruction of operation can be passed.
[0069] As shown in drawing 3 , the mobile robot 1 concerning this example interconnects with the external host computer 100 via the network. Or the means of communications of radio (for example, bluetooth and .11B short distance wireless data transmission) or others may connect in the host computer 100.
[0070] On a host computer 100, a predetermined software execution environment is built, under this environment, an authoring tool can be started and a robot's 1 operating sequence can be created comparatively easily and efficiently by interactive processing. About the detail of an authoring tool, it mentions later.
[0071] In drawing 4 , the example of a hardware configuration of a host computer 100 is illustrated typically. Hereafter, each part in a computer 100 is explained.
[0072] CPU (Central Processing Unit)101 which is the Maine controller of a system 100 performs various kinds of applications under control of an operating system OS. Although OS offers the GUI (Graphical User Interface) environment more preferably, it is good at Windows 98 [UNIX (trademark) or] of U.S. Microsoft/NT, for example.
[0073] CPU101 interconnects with other equipments (after-mentioned) by bus 107 as illustration. The memory address or I/O Address of a proper is given to each device on a bus 107, respectively, and device access is possible for CPU101 by these addresses. An example of a bus 107 is a PCI (Peripheral Component Interconnect) bus.
[0074] Memory 102 is storage used since the program code performed in a processor 101 is stored or the activity data under activation are stored temporarily. Please understand the memory 102 shown in this drawing to be a thing containing both un-volatilizing and volatilization memory.
[0075] The display controller 103 is an exclusive controller for actually processing the drawing instruction which a processor 101 publishes, for example, supports a bit map drawing function equivalent to SVGA (Super Video Graphic Array) or XGA (eXtended Graphic Array). Once it is written in a frame buffer (not shown), the screen output of the drawing data processed in the display controller 103 is carried out at a display 111. Indicating equipments 111 are for example, a CRT (Cathode Ray Tube) display, a liquid crystal display display (Liquid Crystal Display), etc.
[0076] The input device interface 104 is equipment for connecting user input devices, such as a keyboard 112 and a mouse 113, to a system 100. The input device interface 104 answers the coordinate directions input through the key input or mouse 113 by the keyboard 112, and generates interruption to a processor 101.
[0077] According to predetermined communications protocols, such as Ethernet, it can connect with networks, such as LAN (Local Area Network), or the network interface 105

can connect a system 100 to short-distance wireless data transmission like bluetooth or .11B. Generally, the network interface 105 is offered with the gestalt of a LAN adapter card, and the PCI bus slot on a mother board (not shown) equips with it, and it is used.

[0078] In the example shown in drawing 3 , although the host computer 100 interconnects with the robot 1 via wireless data transmission or a network, of course, both may be connected by other means of communications.

[0079] Moreover, on the network, two or more host computers (not shown) are connected in the transparent condition, and the distributed computing environment is built. Distribution of a software program, data contents, etc. is performed on a network. For example, the authoring tool concerning this example, the operating-sequence program for robots created and edited by this authoring tool can be distributed via a network. Moreover, distribution service of such a program/data may be offered the charge or for nothing.

[0080] The external instrument interface 106 is equipment for connecting external devices, such as a hard disk drive (HDD) 114 and the media drive 115, to a system 100. The external instrument interface 106 is based on interface specification, such as IDE (Integrated Drive Electronics) and SCSI (Small Computer System Interface).

[0081] HDD114 is the external storage which carried the magnetic disk as storage support fixed (common knowledge), and excels other external storage in points, such as memory capacity and a data transfer rate. It calls it "install" to the system of a program to place on HDD116 in the condition that a software program can be performed. Usually, the program code of the operating system which a processor 511 should perform, an application program, a device driver, etc. are stored in HDD114 in un-volatilizing. For example, creation and the edited operating-sequence program for robots are installable on HDD114 using the authoring tool concerning this example, and this authoring tool.

[0082] Moreover, the media drive 115 is equipment for loading with portable mold media, such as CD (Compact Disc), and MO (Magneto-Optical disc), DVD (Digital Versatile Disc), and accessing a data-logging side. Portable mold media are used in order to mainly move backing up a software program, a data file, etc. as data of a computer-readable format, and these between systems (sale, circulation, and distribution are included). For example, these portable mold media can be used, and the authoring tool concerning this example, the operating-sequence program for robots created using this authoring tool can be physically circulated and distributed between devices.

[0083] In addition, an example of the host computer 100 as shown in drawing 4 is the compatible machine or succeeding machine of personal computer PC/AT (Personal Computer/Advanced Technology) of U.S. IBM. Of course, it is also possible to apply the computing system equipped with other architecture to the host computer 100 concerning this example.

[0084] In this example, creation and edit of the control program of operation which consists of a series of command/data which describe a robot's 1 predetermined pattern of operation are done using the authoring tool started on the host computer 100. Moreover, creation and the edited control program of operation are transmitted to a robot 1 side using radio means, such as bluetooth and .11B, using this authoring tool, and collaboration-actuation with a host computer 100 and a robot 1 performs debugging processing. That is, the authoring system for a robot's 1 control program of operation is built by induction-association of a host computer 100 and a robot 1.

[0085] In drawing 5 , the whole authoring system configuration is illustrated typically.
[0086] In a host computer 100 side, a user can use the GUI (Graphical User Interface) screen which an authoring tool offers, and can create and edit the scenario of a convention of a robot 1 by mouse actuation. About the actuation on this screen, it mentions later in the GUI screen list for scenario creation. Or a user can use a text editor etc., and can create and edit a robot's 1 control program of operation in a script format (for example, high level language formats, such as C).
[0087] An authoring tool changes the scenario which the user created and edited on the GUI screen, and the control program of the script format created and edited on the text editor of operation into a mnemonic code like the assembler called "RCODE".
[0088] RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, and since it also has fundamental control structures, such as IF and GO, it can be used also as a minimum level script language for robot controls. However, about the detail of RCODE, it explains in detail behind.
[0089] The RCODE actuation control program created and edited on the host computer 100 is movable to a robot 1 side using media, such as a memory stick. Moreover, at the time of debugging of a RCODE actuation control program, a RCODE program is taken out for every line, and it enciphers, and transmits to a robot 1 side serially using radio means, such as bluetooth and .11B.
[0090] On the other hand, in the robot 1 side, it has an interpreter/debugger, middleware, the driver, and the operating system (OS) as activation and the debugging environment of the control program of operation described by RCODE etc.
[0091] An interpreter is a high-level-language program which reads at a time the program of one line described in the RCODE format, interprets it, and performs it. However, in the time of debugging etc., when a RCODE program is transmitted in the format enciphered from the host computer 100 side, once an interpreter decrypts this, it performs interpretation and activation.
[0092] A debugger discovers the error in a RCODE program (bug), and is a program which supports the activity which corrects. That is, activation can be stopped in the line which specified the program,.or the memory at that time and the contents of the variable can be referred to.
[0093] Middleware is a processing engine which performs control of leg formula actuation of the walk and others in a robot 1 etc., recognition processing of the input image from a camera 15, recognition processing of the voice input from a microphone 16, etc.
[0094] A driver is a program code for operating hardware of each joint actuator or others.
[0095] In this example, the driver is mounted in the middleware list by the object-oriented program. The software based on object-oriented is fundamentally treated in the module unit of the "object" which made the processing procedure over data and its data unify. Moreover, if needed, two or more objects are created or one software is completed by combining. Generally, according to object-oriented, it is thought that the efficiency of development and maintenance of software is increased.
[0096] An operating system (OS) performs control about management of the data communication between these objects, and other program executions. OS is also mounted by the object-oriented program.
[0097] B. The GUI screen which the authoring tool concerning this example offers by the

creation / edit following ** of the program for robots of operation using an authoring tool for a user to create and edit a robot's 1 scenario of operation is explained. The configuration of this GUI screen is illustrated in drawing 6 .

[0098] As shown in this drawing, this GUI screen includes the edit window which performs an editing task by making click by the mouse, and drag-and-drop actuation into the keynote, and the components window which components-izes operations sequence frequently used in an editing task, and offers it.

[0099] As for the edit window, the title bar, the menu bar, the tool bar, and the edit field are arranged in order [upper case / maximum].

[0100] A menu bar is the field where the user arranged the list of selectable processing in the longitudinal direction by click actuation, and each menu item "a file (F)", "edit (E)", "a display (V)", "insertion (I)", and "a group (G)" is prepared in this example. If a certain menu item is chosen, the pull down menu which corresponds further will appear.

[0101] In drawing 7 , the pull down menu about a menu item "a file" is illustrated. As shown in this drawing, in this pull down menu, each sub menu "new creation", "it opening", "overwriting", "naming and saving an identifier", "creation of a memory stick image", "a transfer of a memory stick image", and "termination" is arranged.

[0102] Selection of "new creation" creates newly the scenario (henceforth a "project") which describes a series of actuation of a robot. When the project under edit already exists on an edit window, the dialog to which the preservation is urged appears. The edit window shown in drawing 6 is an initial screen when creating a new project, and the Maine group who consists only of a "START" box and "END" box which corresponds to the head and the tail end of a project, respectively is displayed on the edit field.

[0103] Selection of a sub menu "it opens" opens the project saved as a file on an edit window. When the project under edit already exists on an edit window, the dialog to which the preservation is urged is displayed. In this example, a project shall be treated as a file with extension "apk".

[0104] Selection of a sub menu "overwriting" saves the project under edit to the file of a same name. The dialog which asks a file name shall not appear at this time.

[0105] Selection of a sub menu "name and save an identifier" displays the dialog which asks a file name. On this dialog, a user can specify the directory name and file name of a preservation place, and a project is saved by the file name to the specified directory.

[0106] Selection of a sub menu "creation of a memory stick image" creates the image file for writing in a memory stick (or storage media equivalent to this). At this example, an image file shall be saved by extension called "aim" to the same directory as the file of the original apk format.

[0107] If a sub menu "a transfer of a memory stick image" is chosen, it will generate to the full path which had the image file on a memory stick specified.

[0108] Selection of a sub menu "termination" ends the authoring tool under activation.

[0109] Moreover, to drawing 8 , the pull down menu about a menu item "a group" is illustrated. As shown in this drawing, in this pull down menu, "new group creation", "deletion of this group", "a group list", a "group layered structure", "all groups being opened", and each sub menu of "closing all groups" are arranged.

[0110] Selection of a sub menu "new group creation" creates a new group.

[0111] Selection of a sub menu "deletion of this group" deletes the group by whom it is indicated by current. However, even if it deletes a group, the group box which is calling

the group is not deleted.
[0112] Selection of a sub menu "a group list" displays a list of a group. The group can be opened in double-clicking a group's displayed identifier.
[0113] Selection of a sub menu "a group layered structure" displays a group's layered structure (call structure). The group can be opened in double-clicking a group's displayed identifier.
[0114] Selection of a sub menu "all groups are opened" opens all groups' window.
[0115] Selection of a sub menu "all groups are closed" closes all groups' window (however, the Maine group removes).
[0116] Moreover, in the example shown in drawing 6 , each tool carbon button the "normal mode", "connection mode", "a box of operation", a "branching box", and a "group box" is arranged in the tool bar.
[0117] If a carbon button "the normal mode" is chosen, mouse actuation will change to the normal mode. For example, the drag actuation to a box functions as migration of a box.
[0118] If a carbon button "connection mode" is chosen, mouse actuation will change in connection mode. That is, the drag actuation to a box functions as connection between boxes. What is necessary is just to perform drag actuation from a connecting agency to the box of a connection place. Moreover, when it is connection from a branching box, the dialog which specifies branch condition is displayed.
[0119] Selection of a carbon button "a box of operation" inserts a box of operation (Action) in a downward edit field.
[0120] Selection of a carbon button "a branching box" inserts a branching (Branch) box in a downward edit field.
[0121] Selection of a carbon button "a group box" inserts a group box in a downward edit field.
[0122] A user can specify processing with the combination of mouse actuation of a click, a drag, etc., mouse actuation, and a key input on the edit field shown in drawing 6 .
[0123] For example, a box can be chosen by the click of the left carbon button of a mouse.
[0124] Moreover, when the left carbon button of a mouse is double-clicked, a property is displayed and/or a group is opened.
[0125] Moreover, when the left carbon button of a mouse is pushed and dragged under the normal mode and a mouse cursor suits on a box, it moves in a box, and a range selection is performed when a mouse cursor is outside a box. Moreover, boxes are connected when the left carbon button of a mouse is pushed and dragged under connection mode, and a mouse cursor is on a box.
[0126] Moreover, if the combination of the depression/drag of the left carbon button of a mouse is performed the bottom of the [Ctrl] key press under the normal mode, when a mouse cursor suits on a box, a box is connected, and range assignment will be performed when a mouse cursor is outside a box. Moreover, if the combination of the depression/drag of the left carbon button of a mouse is performed the bottom of the [Ctrl] key press under connection mode, when a mouse cursor is on a box, it will move in a box.
[0127] Moreover, an edit window will be resized if the combination of the left carbon button depression / drag of a mouse is performed the bottom of the [Shift] key press.

[0128] Moreover, a click of the right carbon button of a mouse displays a context menu.

[0129] In addition, drag actuation of a mouse is canceled by the depression of the [Esc] key.

[0130] Subsequently, the basic operation procedure on the GUI screen which an authoring tool as shown in drawing 6 offers is explained.

[0131] First, a new project is created by choosing the sub menu [new creation] of a menu [a file]. As the screen of a new project is shown in drawing 6 , the Maine group who consists only of a "START" box and "END" box which corresponds to the head and the tail end of a project, respectively is displayed on the edit field.

[0132] Subsequently, the box which corresponds on an edit window can be arranged by choosing a menu [insertion] or the carbon button in a tool bar [A], [B], and [G]. The box once arranged on an edit window can move an installation location by mouse actuation.

[0133] In drawing 9 , signs that a box [A] of operation and one branching box [B] are arranged each on the edit window are illustrated. Although "Box002" is given to the default identifier "Box001" list, a user can change this into each box later.

[0134] Subsequently, mouse actuation is switched to connection mode by choosing the sub menu [mouse:connection mode] of a menu [edit], or choosing the [connection mode] carbon button in a tool bar. And the boxes in the context on scenario advance are connected by mouse actuation.

[0135] In drawing 10 , signs that the box [A] of operation and the branching box [B] were connected to the [START] box, the box [A] of operation, and the list are illustrated on the edit window shown in drawing 9 .

[0136] Since the dialog (not shown) for specifying the branching surface opens automatically about the connection from a branching box [B] in case boxes are connected, the user should just specify conditions according to the directions in a dialog. However, connection is not made, unless criteria specification of a branching box shall not be made to deferment rather than the time of connection with the next box and it specifies conditions completely.

[0137] Moreover, the dialog for specifying the detail of processing actuation of the box concerned opens a user in double-clicking the left carbon button of a mouse on a certain box. Moreover, when a group box is double-clicked, the group opens as another edit window (after-mentioned).

[0138] Processing actuation of each box is described by this example by mnemonic code like an assembler called RCODE. Although RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, about the detail of RCODE, it explains in detail behind.

[0139] In drawing 11 , the dialog for specifying the detail of a box of operation is illustrated. This dialog can be called by double-clicking a desired box [A] of operation on an edit window as shown in drawing 9 or drawing 10 .

[0140] A user can specify the identifier of a box of operation by inputting a character string into the identifier field on a dialog as shown in drawing 11 . Moreover, the comment about this box of operation can be filled in on a comment field.

[0141] Whether the direct input of the command name and command parameter of RCODE is carried out into combo boxes, such as Action, Part, Sound, and Volume#, furthermore, or the right end of this box By choosing a desired command or a desired command parameter from the list box (not shown) which pushes a carbon button and appears, one line, i.e., the RCODE command for one step, can be set up.

[0142] If the RCODE command for one step is set up using combo boxes, such as these Action(s), Part, Sound, and Volume#, and a carbon button [additional (A)] is clicked further, sequential registration will be carried out on a command list.

[0143] Moreover, it is on this command list, and after choosing a predetermined line, by clicking [modification (M)] carbon button, the line concerned serves as a candidate for modification, and those contents of a setting are displayed on each combo box, such as Action, Part, Sound, and Volume#. Moreover, it is on this command list, and after choosing a predetermined line, the line concerned can be deleted from a command list by clicking [deletion (D)] carbon button.

[0144] And if assignment of the detail of a box of operation is completed on this dialog, while the contents of assignment which minded the screen by clicking the closed [(C]) carbon button will be registered as a manipulation routine described by RCODE, a dialog closes.

[0145] Moreover, to drawing 12 , the dialog for specifying the detail of a branching box is illustrated. This dialog can be called by double-clicking a desired branching box [B] on an edit window as shown in drawing 9 or drawing 10 .

[0146] A user can specify the identifier of a branching box by inputting a character string into the identifier field on a dialog as shown in drawing 12 . In the example of illustration, "mode branching" is filled in as an identifier of the branching box concerned. Moreover, the comment about this branching box can be filled in on a comment field.

[0147] Whether the direct input of the command name and command parameter of RCODE is carried out into combo boxes, such as Type and Variable, furthermore, or the right end of this box By choosing a desired command or a desired command parameter from the list box (not shown) which pushes a carbon button and appears, the RCODE command which describes the conditional judgment of the branching box concerned can be set up.

[0148] If the RCODE command of ********** is set up using combo boxes, such as these Type(s) and Variable, and a carbon button [additional (A)] is clicked further, sequential registration will be carried out on a command list.

[0149] Moreover, it is on this command list, and after choosing a predetermined line, by clicking [modification (M)] carbon button, the line concerned serves as a candidate for modification, and those contents of a setting are displayed on each combo box, such as Type and Variable. Moreover, it is on this command list, and after choosing a predetermined line, the line concerned can be deleted from a command list by clicking [deletion (D)] carbon button.

[0150] And if assignment of the detail of a branching box is completed on this dialog, while the contents of assignment will be registered by clicking the closed [(C]) carbon button as a manipulation routine described by RCODE, a dialog closes.

[0151] In addition, there is not necessarily no need that a user sets up the detail of each box himself using a dialog as shown in drawing 11 or drawing 12 . Since a box of operation, a branching box, etc. which completed detailed assignment beforehand components-ize in the components window on the GUI screen which an authoring tool prepares (for example, refer to drawing 6) and are prepared for it, a user can use the components concerned as it is as a part of program by drags and drops a desired components box from a components window to an edit window.

[0152] Moreover, on the GUI screen (refer to drawing 6) which an authoring tool offers, if the right carbon button of a mouse is clicked on a certain box, "Contextual Menu" (not shown) will appear. [Deletion of a box] and [deletion of connection] can be performed from this Contextual Menu.

[0153] However, on a GUI screen, the specific line which connects between boxes cannot be specified, and the deletion cannot be directed directly, but Contextual Menu must be used for deletion of a line. In the case of a branching box, the dialog (not shown) containing the list box for choosing which connection is deleted appears. A user should just click [deletion] carbon button from which connection to delete out of this list was chosen.

[0154] A user can create the program which described, a scenario, i.e., the operations sequence, to which a robot 1 performs a soccer game, using a GUI screen of an authoring tool which was mentioned above. Such a scenario is described by the box configuration as shown in drawing 13 .

[0155] In the example shown in drawing 13 , after performing a robot's 1 initial setting from the contents specified in the box [initialization] of operation, according to a comparison result with the conditions specified with the branching box [mode branching], a robot 1 performs one mode of operation among [fall recovery], the [search mode], and a [trace mode].

[0156] Here, each of [fall recovery], the [search mode], and a [trace mode] is described as a group box. As mentioned above, when a group box is double-clicked on an edit window, the group opens as another edit window. In drawing 14 , the edit window of the search mode which double-clicks the [search mode] and appears is illustrated. Moreover, to drawing 15 , the edit window of a trace mode which double-clicks a [trace mode] and appears is illustrated.

[0157] By this example, being described by mnemonic code like an assembler called RCODE already stated processing actuation of a robot 1. Here, the detail of RCODE is explained.

[0158] RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, and since it also has fundamental control structures, such as IF and GO, it can be used also as a minimum level script language for robot controls.

[0159] In RCODE, the word of all capital letters is reserved for the constant identifier etc., and cannot be used for the variable name which a user prepares.

[0160] RCODE is equipped with the operator (operator) for specifying the action which should be performed for control, a jump, branching, substitution, the synchronization, the function, etc. Below, the list of RCODE operators is shown.

[0161]

[Equation 1]

<<control>>

EDIT It shifts to the edit mode (transfer of the code which lets serial Rhine pass).

RUN Activation is started.

END The /activation which ends the edit mode is ended.

LOAD: <filename> SAVEwhich reads the code on a memory stick on memory : <filename> <<jump>> which writes in the code on memory on a memory stick

: <label> Label definition. <label> is one or more numeric values (0 expresses the following line).

GO: <lebel> It jumps on a label.

<<branching>>

IF: <op>:<var1>:<const>|<var2>: [<then_label> : <else_label>] A variable 1 <var1>, a constant <const>, or a variable 2 <var2> is compared, and it branches.

<op> " -- = -- " -- " -- < -- > -- " -- " -- < -- " -- " -- < -- = -- " -- " -- > -- " -- " -- > -- = -- " -- six -- a kind -- it is .

SWITCH: <var> The value of a variable is set up as a context value.

CSET: <op>:<var1>|<const1>:<var2>|<const2>:<var3>| <const3> The <op> operator compares a value 1 and a value 2, and if the result is truth, a value 3 will be set up as a context value.

When a CSET instruction appears continuously, the CSET instruction after the CSET instruction which became truly serves as No Operation treatment (****** instruction).

CASE: <const>: <RCODE command> <RCODE command> will be performed if a context value is equal to <const>.

<<substitution>>

LET: <var1>:<const>| <var2> <var1> <- <const> | <var2> Substitution

SET:<var1>:<const>| of the value to a variable <var2> <var1> <- <const> | <var2> A special function works depending on a variable. (It is not mere substitution)

GET: <var1> The contents of <var1> are displayed on a console (for debugging).

<<operation>>

ADD: <var1>:<const>| <var2> <var1><-<var1>+<const>| <var2> Addition

SUB:<var1>:<const>| <var2> <var1> <-<var1>-<const> | <var2> Subtraction

MUL:<var1>:<const>| <var2> <var1><-<var1>*<const>| <var2> Multiplication

DIV:<var1>:<const>| <var2> <var1><-<var1>/<const>| <var2> Division

MOD:<var1>:<const>| <var2> <var1><-<var1>%<const>| <var2> Remainder

AND:<var1>:<const>| <var2> <var1><-<var1>&<const>| <var2> AND

IOR:<var1>:<const>| <var2> <var1><-<var1>|<const>| <var2> OR

XOR:<var1>:<const>| <var2> <var1><-<var1>^<const>| <var2> Exclusive-OR

RND:<var1>:<from>: <to> <var1> <- range <from>- <to> Uniform-random-number value <<synchronous>>

WAIT It waits for termination (reproductive termination) of the actuation performed immediately before.

WAIT: <ms> It waits only for the ms made into the argument. <ms> = 1-30000. (Notes: Resolution 32 [ms])

WAIT:SOUND It waits for termination of sound playback (it can use also for the judgment under sound playback independently).

WAIT:LIGHT It waits for termination of photoregeneration (it can use also for the judgment under photoregeneration independently).

<<function>>

CALL: <label> <label> A subroutine is called.

* A nest is a maximum of 16 pieces. Suppose parameter delivery that a variable is substituted.

RETURN Return from a subroutine .

* Since the stack pointer did not check, it needs to be cautious of the adjustment of a nest.

[0162] The command which controls "a motion" of a robot 1 among the commands of

RCODE is divided roughly into the following five basic forms. Namely, [0163] (1) POSE It is made to take a certain posture (initiation posture of a motion).
(2) MOVE A certain motion is carried out. OMNE/OMTE/OMLE2_S/OMSE Correspondence.
(3) PLAY A certain motion is reproduced. OAllReplay/OHeadReplay/OLegsReplay/OTailReplay It corresponds.
(4) STOP Halt of operation (since the current line ends the motion which is).
(5) QUIT Emergency shut down of operation (immediately).
[0164] MOVE and PLAY can be said to be being only the difference in whether inclusion actuation is performed or actuation defined as the "*.mtn" file on a memory stick is performed, and being the same function at the point of "carrying out a certain actuation." MOVE Although it is not customizable since actuation is inclusion, fine parameter assignment etc. can be performed. On the other hand, it is substitutable, although parameter assignment cannot be performed since actuation of PLAY is playback of a "*.mtn" file.
[0165] Below, the list of the RCODE commands is shown.
[0166]
[Equation 2]
SET:ROBOT:POWER: One Power-source ONSET:ROBOT:POWER: Zero Power-source OFFGET:ROBOT All variable discharge GET:ROBOT: <var> Variable Acquisition SET:ROBOT:<var>: <value> Variable Setting POSE:ROBOT: <motion_name> To the initiation pause of motion POSE:LEGS: <motion_name> To the initiation pause of motion POSE:HEAD: <motion_name> It is POSE:TAIL: to the initiation pause of motion <motion_name>. It is PLAY:ROBOT: to the initiation pause of motion <motion_name>. motion playback PLAY:LEGS: <motion_name> motion playback PLAY:HEAD: <motion_name> motion playback PLAY:TAIL: <motion_name> motion playback PLAY:SOUND:<sound_name>: <volume> Sound playback volume =:[0-100PLAY:LIGHT] <pattern_name>: <times> Photoregeneration (LED pattern)
times = 0 - 16 MOVE:ROBOT (0:Default 16:Loop) : <motion_name> OMLE inclusion actuation playbacks (fall return etc.)
MOVE:LEGS:WALK: <style1>:<dir>: <times> Walk
MOVE:LEGS:STEP:<style1>:<dir>: <times> Walk (StepWalk)
MOVE:LEGS:KICK: <style2>: <deg> A kick 12 (R-Turn) style1 = 0-10, 13 (L-Turn) style 2 = 14 (R-Kick), 15 (L-Kick) dir = [1-6] times = [0-9999] deg =-90 - 90 MOVE:HEAD:HOME head home-position MOVE:HEAD:ABS:<tilt>:<pan>:<roll>: <time> Head Absolutely Impaction efficiency MOVE:HEAD:REL:<tilt>:<pan>:<roll>: <time> head Relative-position migration MOVE:HEAD:C-TRACKING Head color tracking MOVE:HEAD:C-TRACKING: <time> Head color tracking tilt =-180 - 180 pan =-180-180 roll =-180-180 time = 0-99999 [ms] MOVE:TAIL:HOME Tail Home-position MOVE:TAIL:ABS: <tilt>:<pan>: <time> Tail absolute impaction efficiency MOVE:TAIL:SWING:<tilt>:<pan>: <time> Tail swing tilt =-90-90 pan =-90-90 time = 0-99999 [ms] whole STOP:ROBOT usual halt STOP:LEGS Limbs usual halt STOP:HEAD Head usual halt STOP:TAIL Tail usual halt STOP:SOUND Sound playback halt STOP:LIGHT Photo-regenerating halt QUIT:ROBOT Whole emergency shut down QUIT:LEGS limbs emergency shut down QUIT:HEAD Head emergency shut

down QUIT:TAIL tail emergency shut down QUIT:SOUND (STOP:SOUND and equivalence) sound playback halt QUIT:LIGHT (STOP:LIGHT and equivalence) Photo-regenerating halt [0167] Moreover, RCODE defines the system variable, in order to direct actuation / control value or to acquire the condition of each part to a robot's 1 each part (read-out of a sensor output value etc.). Below, the list of the system variables of RCODE is shown.

[0168]

[Equation 3]

Power A power source 0:-1:ONHead_tilt 1st joint include-angle [**] Head_pan of a head 2nd joint include-angle [**] Head_roll of a head 3rd joint include-angle [**] Head_mouth of a head Head temporomandibular-joint include-angle [**] Tail_1 1st joint include-angle [**] Tail[of a tail]_2 2nd joint include-angle [**] Leg_RF[of a tail]_1 1st joint include-angle [**] Leg_RF[of the right nose gear]_2 2nd joint include-angle [**] Leg_RF[of the right nose gear]_3 3rd joint include-angle [**] Leg_LF[of the right nose gear]_1 1st joint include-angle [**] Leg_LF[of the left nose gear]_2 2nd joint include-angle [**] Leg_LF[of the left nose gear]_3 3rd joint include-angle [**] Leg_RR[of the left nose gear]_1 the right hind leg 1st -- joint include-angle [**] Leg_RR_2 2nd joint include-angle [**] Leg_RR[of right hind legs]_3 3rd joint include-angle [**] Leg_LR[of right hind legs]_1 1st joint include-angle [**] Leg_LR[of left hind legs]_2 2nd joint include-angle [**] Leg_LR[of left hind legs]_3 3rd joint include-angle [**] Head_sw of left hind legs Head pressure sensor [10^-3Pa] Distance Head obstruction sensor [mm] Leg_RF_sw Right nose-gear **** sensor [On:-1 Off:0] Leg_LF_sw Left nose-gear **** sensor [On:-1 Off:0] Leg_RR_sw Right hind leg **** sensor [On:-1 Off:0] Leg_LR_sw Left hind leg **** sensor [On:-1 Off:0] Gsensor_status G sensor status 16-bit flag Gsensor_roll G sensor roll angle [**] Gsensor_pitch G sensor pitch angle [**] Gsensor_yaw G sensor yaw angle [**] Cdt_npixel Number [Pixels] Psd_status of color-sensor pixels Obstruction sensor situation Psd_range Distance Touch_head to an obstruction Head touch sensor Touch_head_time Head touch sensor depression time amount Touch_head_press Head touch sensor mean pressure [unit -- unknown --]Touch_RF right nose-gear **** sensor ON->OFF/OFF->ON detection Touch_LF Left nose-gear **** sensor ON->OFF/OFF->ON detection Touch_RR right hind leg **** sensor ON->OFF/OFF->ON detection Touch_LR Left hind leg **** sensor ON->OFF/OFF->ON detection Tone_num Tone detection Sound number Tone_level Loudness level Tone_dir Detection direction Melody_id Melody detection Melody IDMelody_num The number of tones (1-3)

Tone1_num Tone first-sound number

Tone1_level Magnitude Tone1_dir of a Tone first sound Tone1 detection direction

Tone2_num Tone second-sound number

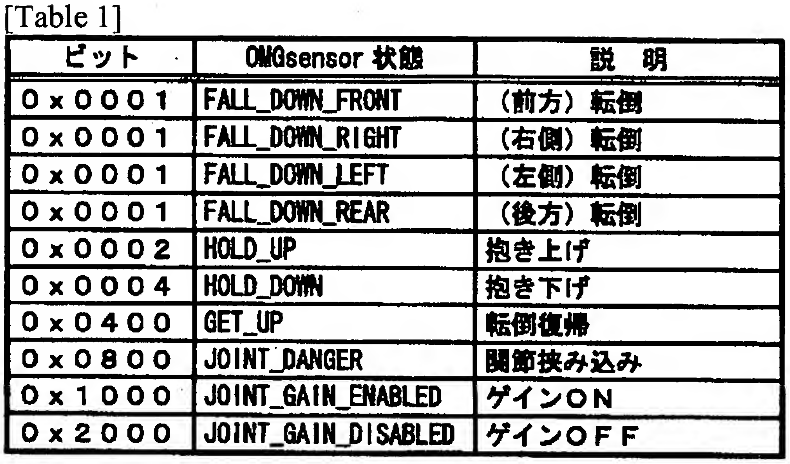
Tone2_level Magnitude Tone2_dir of a Tone second sound Tone2 detection direction

Tone3_num Tone3 sound number

Tone3_level Tone3 loudness-level Tone3_dir Tone3 detection direction Sound_status Sound detection Status Sound_num Sound number

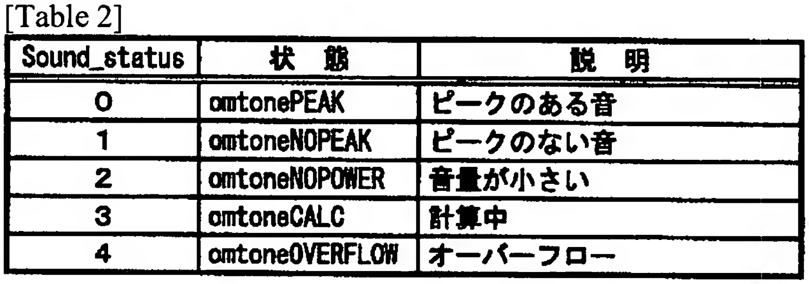
Sound_level Loudness level Sound_dir Detection direction Sound_busy Under 1:sound playback 0-:-sound-less Light_busy Under 1:Light playback [with no 0:Light -- 0169] Although Gsensor_status consists of 16-bit flags among the above-mentioned system variables, the semantics of each bit is as follows.

[0170]
[Table 1]



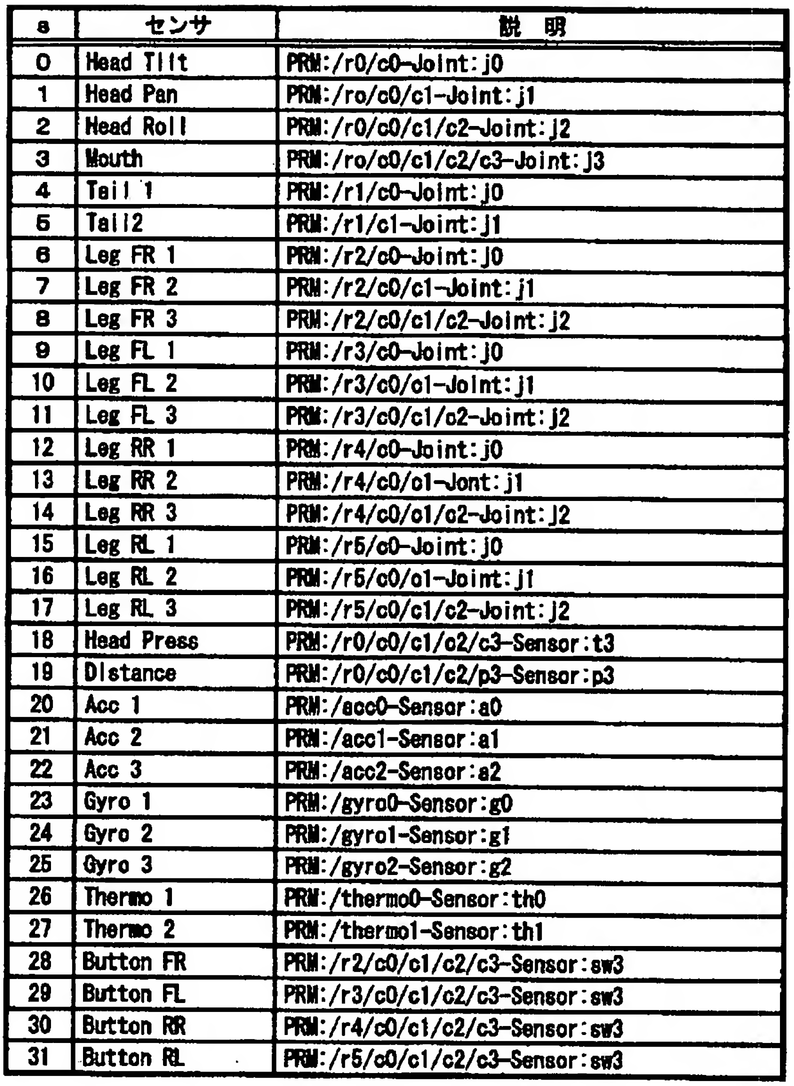
ビット	OMGsensor 状態	説 明
０ｘ０００１	FALL_DOWN_FRONT	（前方）転倒
０ｘ０００１	FALL_DOWN_RIGHT	（右側）転倒
０ｘ０００１	FALL_DOWN_LEFT	（左側）転倒
０ｘ０００１	FALL_DOWN_REAR	（後方）転倒
０ｘ０００２	HOLD_UP	抱き上げ
０ｘ０００４	HOLD_DOWN	抱き下げ
０ｘ０４００	GET_UP	転倒復帰
０ｘ０８００	JOINT_DANGER	関節挟み込み
０ｘ１０００	JOINT_GAIN_ENABLED	ゲインＯＮ
０ｘ２０００	JOINT_GAIN_DISABLED	ゲインＯＦＦ

[0171] Moreover, system variable Sound_status expresses the voice input condition to a robot 1 with the following values.
[0172]
[Table 2]



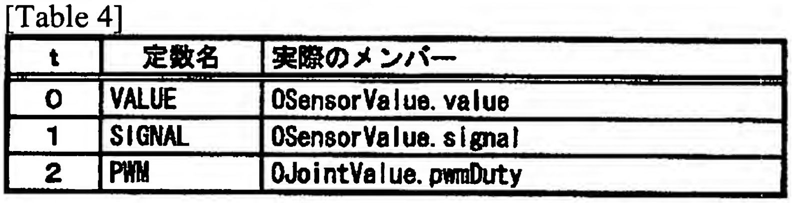
Sound_status	状 態	説 明
0	omtonePEAK	ピークのある音
1	omtoneNOPEAK	ピークのない音
2	omtoneNOPOWER	音量が小さい
3	omtoneCALC	計算中
4	omtoneOVERFLOW	オーバーフロー

[0173] Moreover, RCODE offers the service which takes the log of the output data of the sensor arranged by at least each part on a robot 1 (it stores temporarily to the predetermined field on memory). That is, while directing initiation of logging by RCODE command "LOG:START", termination of logging can be directed by RCODE command "LOG:END". Moreover, the log number of cases memorized by variable "Log_n" is expressed.
[0174] Furthermore, a log can be saved by the specified file name by RCODE command "LOG:PCFILE:<file>" at the file on a host system 100. Similarly, a log can be saved by the specified file name by RCODE command "LOG:MSFILE:<file>" at the file on a memory stick.
[0175] Moreover, it may be made to carry out a direct printout by the printer serially, without saving the output data of each sensor at memory or a file. in this case, RCODE command" -- LOG:PRINT : -- -- < -- s -- > -- : -- < -- t -- > -- : -- < -- i -- > -- " -- an output form -- it can specify . However, in s, (referring to [table 3]) and t shall show the class of value, and (referring to [table 4]) and i shall show an index (0 - Log_n -1) for a sensor number, respectively. Moreover, line feed can be directed by RCODE command "LOG:PRINT:CRLF", and file closing can be directed by RCODE command "LOG:PRINT:EOF", respectively.
[0176]
[Table 3]



s	センサ	説　明
0	Head Tilt	PRM:/r0/c0-Joint:j0
1	Head Pan	PRM:/ro/c0/c1-Joint:j1
2	Head Roll	PRM:/r0/c0/c1/c2-Joint:j2
3	Mouth	PRM:/ro/c0/c1/c2/c3-Joint:j3
4	Tail 1	PRM:/r1/c0-Joint:j0
5	Tail2	PRM:/r1/c1-Joint:j1
6	Leg FR 1	PRM:/r2/c0-Joint:j0
7	Leg FR 2	PRM:/r2/c0/c1-Joint:j1
8	Leg FR 3	PRM:/r2/c0/c1/c2-Joint:j2
9	Leg FL 1	PRM:/r3/c0-Joint:j0
10	Leg FL 2	PRM:/r3/c0/o1-Joint:j1
11	Leg FL 3	PRM:/r3/c0/c1/o2-Joint:j2
12	Leg RR 1	PRM:/r4/c0-Joint:j0
13	Leg RR 2	PRM:/r4/c0/o1-Jont:j1
14	Leg RR 3	PRM:/r4/c0/o1/o2-Joint:j2
15	Leg RL 1	PRM:/r5/c0-Joint:j0
16	Leg RL 2	PRM:/r5/c0/o1-Joint:j1
17	Leg RL 3	PRM:/r5/c0/c1/c2-Joint:j2
18	Head Press	PRM:/r0/c0/c1/c2/c3-Sensor:t3
19	Distance	PRM:/r0/c0/c1/c2/p3-Sensor:p3
20	Acc 1	PRM:/acc0-Sensor:a0
21	Acc 2	PRM:/acc1-Sensor:a1
22	Acc 3	PRM:/acc2-Sensor:a2
23	Gyro 1	PRM:/gyro0-Sensor:g0
24	Gyro 2	PRM:/gyro1-Sensor:g1
25	Gyro 3	PRM:/gyro2-Sensor:g2
26	Thermo 1	PRM:/thermo0-Sensor:th0
27	Thermo 2	PRM:/thermo1-Sensor:th1
28	Button FR	PRM:/r2/c0/c1/c2/c3-Sensor:sw3
29	Button FL	PRM:/r3/c0/c1/c2/c3-Sensor:sw3
30	Button RR	PRM:/r4/c0/c1/c2/c3-Sensor:sw3
31	Button RL	PRM:/r5/c0/c1/c2/c3-Sensor:sw3

[0177]
[Table 4]



t	定数名	実際のメンバー
0	VALUE	OSensorValue.value
1	SIGNAL	OSensorValue.signal
2	PWM	OJointValue.pwmDuty

[0178] If the authoring tool concerning this example is used and the scenario for a robot's 1 motion control is created and edited as already explained, automatic conversion will be carried out at RCODE for realizing the edited scenario. For example, the control program of the scenario for soccer games with the main routine generated on the GUI screen as shown in drawing 13 of operation is described as follows using RCODE as shown below.
[0179]
[Equation 4]
SETV:ROBOT:POWER: 1 SET:Trace:1 SET:mode:0// 0: Search mode 1: Run after and it is mode SET:head:0//. Counter SET:lost:0//for acting as the guru guru of the head by

search mode The count which missed the ball : 100//MainLoopSET:stat:Gsensor_statusAND:stat:1IF:=:stat:1:9000//Did it slip down? IF: =:mode:0:1000 It IF:=:mode:1:2000//Runs after to//search mode, and is GO:100:1000//mode:0 to the mode. Carrying out search mode SET:mode:0 MOVE:LEGS:STEP:RIGHT_TURN:0:4// right-handed rotation A ball SWITCH:head// head to look for Guru guru ****** processing CASE: Zero: MOVE: HEAD: ABS: - 15: Zero: Zero: 500CASE: One: MOVE: HEAD: ABS: - 15: - 40: Zero: 500CASE: Two: MOVE: HEAD: ABS: - 15: - 80: Zero: 500CASE: Three: MOVE: HEAD: ABS: - 45: - 40: Zero: 500CASE: Four: MOVE: HEAD: ABS: - 45: Zero: Zero: 500CASE: Five: MOVE: HEAD: ABS: - 45: 40:0:500CASE:6:MOVE:HEAD:ABS:-45:80:0:500CASE:7:MOVE:HEAD:ABS:-15:40:0:500CASE:8:MOVE:HEAD:ABS:-15:0:0:500 ADD:head:1 //head ** Increment MOD:head:9WAITIF:<:Cdt_npixel:32:100//Ball vanity ******** It runs after and is the mode. :2000//Run after. Mode SET:mode:1IF:<:Cdt_npixel:32:1000 Are they // ball vanity ******?
MOVE:HEAD:C-TRACKING: 100 //Color trace initiation IF:>:Head_tilt:-58:2300 //It is distance presumption IF:>:Head_pan:0:2210:2220 at the include angle of the head. //It will kick, if near!
: 2210 MOVE:HEAD:HOME//So that its AGO may not be kicked (^^;) as unnatural as **** -- MOVE:LEGS:KICK:LEFT_KICK:0 MOVE:LEGS:STEP:SLOW:FORWARD:1GO:2900:2220 MOVE:HEAD:HOME//One's AGO it does not kick -- as -- (^^;). as unnatural as **** -- MOVE:LEGS:KICK:RIGHT_KICK:0 MOVE:LEGS:STEP:SLOW:FORWARD:1GO:2900:2300//With the include angle of the head CSETapproaching a ball : >: Head_pan: 60: 1CSET: >: Head_pan: 45: 2CSET(s): >: Head_pan: 15: 3CSET(s): <: Head_pan: - 60: 4CSET(s): <: Head_pan: - 45: 5CSET(s): <: Head_pan: - 15: 6CSET(s): = : Zero: Zero: 0CASE: Zero: MOVE: LEGS: STEP: SLOW: FORWARD: 4CASE: One: MOVE: LEGS: STEP: SLOW: FORWARD: 4CASE: Two: MOVE: LEGS: STEP: LEFT_TURN: Zero: 4CASE: Three: MOVE: LEGS: STEP: SLOW: LEFT: 4CASE: Four: MOVE: LEGS: STEP: SLOW: LEFTFORWARD: 4CASE: Five: MOVE: LEGS:STEP:RIGHT_TURN:0:4CASE:6:MOVE:LEGS:STEP:SLOW:RIGHT:4CASE:7 :MOVE:LEGS:STEP:SLOW:RIGHTFORWARD:4:2900WAITGO:100:9000 // Fall recovery QUIT:ROBOTMOVE:ROBOT:ReactiveGUWAITGO:100[0180] Moreover, as explained referring to drawing 5 , in the authoring system concerning this example, a high level language like C can be used, for example, and a control program of operation can be created and edited in a script format. The script which realizes a function and operations sequence equivalent to the RCODE program shown with the above-mentioned [-four number] is shown below. However, the script shall be described according to the syntax of C.

[0181]

[Equation 5]

// // Soccer game////#define ROBOT_NEUTRAL described in the script format (0) // Neutral//#define ROBOT_FOUND (1) // Ball **********//#define ROBOT_LOST (2) // //#define ROBOT_FALL which is not from ball **** (99)//ROBOT fall// Pink//#define PinkBall (32) // System variable extern int Head_tilt, Head_roll;extern int

Head_pan;extern int Cdt_npixel;extern int Gsensor_status;// Whether the ball was found or it missed int bBall = 0, bBoil;// Current condition int nSoccer = 0;int nLost = 0; // A ball Number-of-steps//when missing The front declaration int of a function SoccerDog(int);void NeutralROBOT();void FindBall();void SearchBall();void RecoveryGU();void NearBall();void KickBall();// Maine void main (void) -- {-- // Soccer flag int bSoccerFlag; int dummy = 0, true = 1; bSoccerFlag = dummy = true; // A power source ON PowerOn(); // A main loop while (bSoccerFlag) -- {-- // A soccer locking dog bSoccerFlag = SoccerDog (1); } //power-source OFF PowerOff(); }// the soccer initiation int SoccerDog (int dummy) -- {-- // Divide processing in the current condition. switch(nSoccer) { case 0 : //ROBOT_NEUTRAL : // neutral condition NeutralROBOT();break;case 1://ROBOT_FOUND : // Found the ball. FindBall(); break; case 2: //ROBOT_LOST: // Missed the ball. SearchBall(); break; case 99: //ROBOT_FALL: // The Aibo fall RecoveryGU(); // Fall return processing } //continuation return 1; the processing void FindBall (void) which finds //FALSE;}// ball and approaches -- {-- int nHead;nHead = Head_tilt; // The Kalat racking RcodeColorTracking(100); // The distance of a ball at an angle of the head Measurement if(nHead > -58) {// A ball is detected. if(Cdt_npixel < 32) {// Missed the ball. nSoccer = 2; //ROBOT_LOST; } else -- {-- // A ball is approached. NearBall(); } } else -- {-- // A ball is kicked. Ki ckBall(); } }// void NearBall (void) approaching a ball -- {-- // Set up the location of a ball at an angle of a neck. if (Head_pan == 0) -- {-- RcodeStepWalk (0, 1, 4); //StepWalk(FORWARD, 4); RcodeWait(); return; } if(Head_pan> 60) { RcodeStepWalk(0, 1, 4);// StepWalk(FORWARD, 4);RcodeWait();return; } if (Head_pan > 45) -- {-- RcodeStepWalk (0, 13, 4); // StepWalk(LEFT_TURN, 4); RcodeWait();return; } if(Head_pan> 15) {RcodeStepWalk (0, 5, 4); // StepWalk (LEFT, 4); RcodeWait(); return; } if (Head_pan <-60) -- {-- RcodeStepWalk (0, 3, 4); // StepWalk(LEFTFORWARD, 4); RcodeWait(); return;} if (Head_pan <-45) -- {-- RcodeStepWalk (0, 1, 4); // StepWalk(RIGHT_TURN, 4); RcodeWait(); return;} if(Head_pan <-15) { RcodeStepWalk (0, 4, 4); //StepWalk(RIGHT, 4); RcodeWait(); return; }// Detect a ball. if {(Cdt_npixel < 32) nSoccer = 2; //ROBOT_LOST:return; } }// void KickBall (void) which kicks a ball -- {-- // Justification of the head RcodeHead(0, 0, 0, 0); // a ball -- right and left of the head -- the guide peg kicked by in which it is is decided if (Head_pan> 0) -- {-- RcodeKick (15 0); // left leg }else -- {-- RcodeKick(14 0); // A right leg } // Since it shifts to a walk, 1 step **** RcodeStepWalk(0, 0, 1); // StepWalk(FORWARD, 1);}// the processing voidSearcBall (void) which looks for a ball -- {-- // Move the head. switch (nLost) -- {-- case 0: RcodeHead (-15, 0, 0, and 500); break; case 1 : RcodeHead(-15, -40, 0, and 500); break; case 2 : RcodeHead(-15, -80, 0, and 500); break; case 3 : RcodeHead (-45, -40, 0, 500); break; case4: RcodeHead(-45, 0, 0, and 500); break; case 5: RcodeHead(-45, 40, 0, and 500); break; case 6 : RcodeHead(-45, 80, 0, and 500); break; case 7: RcodeHead (-15, 40, 0, and 500); break; case 8: RcodeHead (-15, 0, 0, and 500); break; } // A setup of the number of steps nLost = % (nLost+1) 9; // Walk on right-hand side. RcodeStepWalk (12, 1, 4); //StepWalk(RIGHT_TURN, 4); // Color detection if -- {(Cdt_npixel < 32) -- nSoccer = 2; //ROBOT_LOST; } else{ // Initialization of nLost nLost = 0; // Processing approaching ROBOT nSoccer = 1;//ROBOT_FOUND; } }// // the processing void NeutralROBOT at the time of a neutral (void) -- {-- // fall detection if {(Gsensor_status & 1) // fall nSoccer = 99; //ROBOT_FALL; return;} // The fall has not been carried out. Ball detection if --

{(Cdt_npixel < 32) -- // Missed the ball. nSoccer = 2; //ROBOT_LOST; } else -- {-- // the ball was found nSoccer = 1; //ROBOT_FOUND; } }// the fall return void RecoveryGU (void) -- {-- // Fall return success.
RcodePlay ("ReactiveGU"); RcodeWait(); // A current condition is made neutral. nSoccer = 0; //ROBOT_NEUTRAL;} [0182] As for the authoring system concerning this example, a user can do automatic conversion of creation and the edited script at the program of a RCODE format using a text editor etc. (above-mentioned). Therefore, automatic conversion of the script shown in [the-five number] is carried out at the program code of a RCODE format as shown in [the-four number] using the authoring tool concerning this example.

[0183] For example, when the host system 100 which edits a processing program of operation using RCODE, and the robot 1 which performs a processing program of operation are connected through wireless data transmission (or means of communications of other wireless or a cable) like bluetooth or .11B, a host system 1 transmits a RCODE program per one line, i.e., 1 step, and by the robot 1 side, an interpreter can interpret this serially and can perform it. Moreover, a host-system 1 side may encipher transmit data at the time of a command transfer.

[0184] The authoring tool concerning this example is preparing the edit window for the editing task of the script described in the RCODE format, and the list for the debugging activity of the real time by the collaboration-actuation by the side of a robot 1. In order to distinguish from the edit window shown in drawing 6 , below, this edit window is made to call it a "script edit window."

[0185] The configuration of a script edit window is illustrated in drawing 16 .

[0186] The script edit window is equipped with seven combo boxes, and can specify a command in a hierarchical menu format. Signs that the command of the 1st line of an edit field is specified are shown by the example shown in this drawing.

[0187] If the [Command] carbon button under a combo box is clicked, the specified command will be transmitted to a robot 1 side. Although the transmittal mode is based on wireless data transmission, such as bluetooth and .11B, they may be other communication modes. Moreover, the command transfer performed here operates to asynchronous with the below-mentioned [Execute].

[0188] Moreover, a click of the [**] carbon button on the left of the [Command] carbon button sticks the specified run string on a downward edit field.

[0189] On the edit field arranged under the [Command] carbon button, the direct edition of the script can be carried out by key input.

[0190] Moreover, each carbon button [Open], [Save], [Send], [Execute], and [Stop] is arranged in the lower part by the pan of an edit field.

[0191] By mouse actuation of the [Open] carbon button, a desired script file can be chosen from the local disk of a host system 100 etc., and it can display on an edit field.

[0192] By mouse actuation of the [Save] carbon button, the script currently displayed on the edit field can be saved at a file.

[0193] By clicking the [Send] carbon button, the script currently displayed on the edit field can be transmitted to a robot 1 side. Although the transmittal mode is based on wireless data transmission, such as bluetooth and .11B, they may be other communication modes. However, a script is not only transmitted and is not necessarily performed as it is on a robot 1. In addition, at the time of termination of [Send] processing, the line "END" is displayed in a console window (not shown).

[0194] By clicking the [Execute] carbon button, the script play initiation transmitted to the robot 1 side can be directed.

[0195] By clicking the [Stop] carbon button, the forcible stop of the script under activation can be directed in a robot 1.

[0196] C. As program execution of operation / debugging **** was carried out, the corresponding command of one line can be transmitted to a robot 1 side by [which are depended on a robot] clicking the [Command] carbon button on a script edit window. Moreover, the script currently opened to the edit field can be transmitted to a robot 1 side by pushing the [Send] carbon button. It is desirable to send out at the time of such a command transfer, once it enciphers command data.

[0197] In the robot 1 side of another side, it has an interpreter/debugger, middleware, the driver, and the operating system (OS) as activation and the debugging environment of the control program of operation described by RCODE etc. (refer to drawing 5).

[0198] An interpreter is a high-level-language program which reads at a time the program of one line described in the RCODE format, interprets it, and performs it. Moreover, in the time of debugging etc., when a RCODE program is transmitted in the format enciphered from the host computer 100 side, once an interpreter decrypts this, it performs interpretation and activation.

[0199] A debugger discovers the error in a RCODE program (bug), and is a program which supports the activity which corrects. That is, activation can be stopped in the line which specified the program, or the memory at that time and the contents of the variable can be referred to.

[0200] Middleware is a processing engine which performs control of leg formula actuation of the walk and others in a robot 1 etc., recognition processing of the input image from a camera 15, recognition processing of the voice input from a microphone 16, etc. Middleware can operate hardware of each joint actuator or others through a driver.

[0201] Middleware generates the information (here, it will be temporarily called "posture transition information") for changing in a target posture or target actuation in a robot's 1 leg formula motion control. That is, based on the action command information supplied from high order programs, such as an interpreter, the posture transition information for making the following posture or actuation (target posture or target actuation) change is generated from a robot's 1 current posture, or actuation. The posture which can change next from a current posture is determined by equipment configurations, such as a configuration of a fuselage, a hand, or a foot, weight, a physical configuration of a robot 1 like the integrated state of each part, a direction at which each joint turns, and an include angle. Middleware generates posture transition information in consideration of such an equipment configuration.

[0202] In this example, middleware has registered beforehand the posture in which a robot 1 can change, and the actuation at the time of changing in the form of a directed graph, changes the action command from a high order program into posture transition information according to a directed graph, and publishes the concrete directions of operation to a driver.

[0203] Generally, a robot 1 may be unable to transit directly according to the contents of a command described by RCODE depending on the current posture. That is, a robot's 1 posture is classified into the posture which can be transited directly from a current posture, and the posture whose **** becomes possible via actuation and the posture which it has not changed directly.

[0204] For example, although it can move to the condition of having turned down hand and foot from the condition of having given up greatly and having lain down, directly when it is robot equipment of a quadrapedalism mold, as a robot 1 shows drawing 1 , at least two steps need to be operated [of starting via the posture which could not transit directly to the posture which stood, but once drew near and turned down hand and foot near the fuselage].

[0205] moreover, if it does not go via two steps of actuation in this way, the posture transition which cannot be performed without a robot 1 falling namely, -- also exists in insurance. For example, the robot 1 of a quadrapedalism mold as shown in drawing 1 may be reversed, if both forefeet tend to be mentioned with the posture which stands and it is going to carry out "Banzai!." Or when the contents which sit down as contents of the command and can do only a posture when a current posture lies down and it is in a posture (recumbent form vigor) of "carry out Bata Bata of the guide peg" have be send and the instruction to which Bata Bata of the guide peg carries out as it is publishes in each Seki nodal drive section, the actuation to which it sits on from recumbent form vigor, and Bata Bata of the transition and the guide peg to a posture carries out will be perform. Consequently, a robot 1 will lose balance and will fall.

[0206] In this example, middleware holds the directed graph connected with the actuation which makes each posture change while the posture and actuation which a robot 1 can take are registered. And it searches for the path to the posture or actuation made into a target from a current posture (that is, directed) on this directed graph, and an instruction which makes it change to the posture or actuation made into a target from a current posture is generated based on that retrieval result.

[0207] In drawing 17 , the example of the directed graph used in this example is illustrated. A directed graph consists of owner ** arcs (arc of operation) which connect between the node which shows the posture which a robot 1 can take, and two nodes (posture) which can be changed. the example shown in drawing 17 -- as a node, i.e., a posture, -- "-- ******" -- "-- lie down -- five kinds of **", "it sitting down", "it leaving", and "a certain **" are included. Moreover, an owner ** arc has the arc of operation which changes from a certain posture to other postures, the arc of operation which returns from other postures to a certain posture, and the self-actuation arc which actuation completes within one node. Moreover, there may be two or more self-actuation arcs about that there are two or more arcs of operation which change from a certain posture to other postures, and a certain posture.

[0208] for example, a robot's 1 present posture -- "-- lie down -- when it is **" and the RCODE command which directs what "is sat down" to an interpreter is issued, it is shown in drawing 17 -- as -- "-- lie down -- since an owner ** arc exists between each node of "sitting down" with **", direct transition is possible. Therefore, middleware should just publish a drive command to a driver according to the directions passed from an interpreter.

[0209] on the other hand, the present posture -- "-- lie down -- the time of being **" -- an interpreter -- receiving -- "a certain **" -- when the RCODE command which directs things is issued, it is shown in drawing 17 -- as -- "-- lie down -- the owner ** arc directly connected from the node **" to the node "a certain **" does not exist. therefore, middleware -- a directed graph top -- a node -- "-- lie down -- it goes to a node "a certain **" from **" -- a node [path planning is carried out and] "it leaves" -- going -- a node -- "-- lie down -- the posture transition plan to change from **" to a node "a certain **" is formed, and drive directions are published to a driver based on this plan.

[0210] If the above is summarized, according to this example, middleware will have managed a robot's 1 posture and transition property of operation, and will offer the mechanism which can absorb the difference between logical posture and directions of operation on RCODE command level, and the posture and the transition property of operation on hardware.

[0211] Therefore, a general user does not need to care about detailed hardware information like the posture and the transition property of operation in the actual robot 1, when editing a robot's 1 scenario of operation or describing a control program of operation in a script format on a GUI screen using an authoring tool. Moreover, since the difference between logical posture and directions of operation on RCODE command level, and the posture and the transition property of operation on hardware is absorbed in middleware level, in it being contrary to a robot's posture and transition property of operation, the object of debugging of a RCODE program does not become.

[0212] It has explained in detail about this invention, referring to a specific example more than [addenda]. However, it is obvious that this contractor can accomplish correction and substitution of this example in the range which does not deviate from the summary of this invention.

[0213] Although explained in detail in this example about the authoring system which mentions as an example the pet mold robot which performs quadrapedalism which imitated the dog, and starts this invention, the summary of this invention is not limited to this. for example, a leg formula mobile robot of 2 pairs of shoes like a humanoid robot -- or please understand enough that this invention is applicable similarly also to migration mold robots other than a leg formula.

[0214] Or it is also possible to apply the authoring system which starts this invention for creation and edit of the operating sequence of the animation using not a physical machinery like a robot but the character generated by computer graphics.

[0215] In short, with the gestalt of instantiation, this invention has been indicated and it should not be interpreted restrictively. In order to judge the summary of this invention, the column of the claim indicated at the beginning should be taken into consideration.

[0216]

[Effect of the Invention] As a full account was given above, according to this invention, the outstanding authoring system and outstanding approach of creating a series of command/data which describe a robot's predetermined pattern of operation can be offered.

[0217] Moreover, according to this invention, the outstanding authoring system and outstanding approach of creating a pattern of operation using the set of the components which specify a robot's operating state can be offered.

[0218] Moreover, according to this invention, the outstanding authoring system and outstanding approach of arranging each part article on a computer display, and creating a pattern of operation can be offered.

[Translation done.]

* NOTICES *
JPO and INPIT are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to the authoring system and approach for creating the data according to a predetermined scenario, and relates to the authoring system and approach of creating a series of command/data which describe a robot's predetermined pattern of operation especially.
[0002] Furthermore, in detail, this invention relates to the authoring system and approach of creating a pattern of operation using the set of the components which specify a robot's operating state, and relates to the authoring system and approach of arranging each part article on a computer display, and creating a pattern of operation especially.

[Translation done.]

* NOTICES *
JPO and INPIT are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] The thing of the machinery which performs movement modeled on actuation of human being using the electric or magnetic operation is called "robot." It is said that a robot's origin of a word originates in ROBOTA (slave machine) of a slab word. Although it was in our country that a robot began to spread from the end of the 1960s, the many were the industrial robots (industrial robot) in works aiming at automation, full automation, etc. of production, such as a manipulator and a carrier robot.
[0004] Recently, the researches and developments about leg formula mobile robots, such as a robot (humanoid robot) of "the human form" which imitated the body mechanism of the animal which performs 2-pair-of-shoes walks in erect posture, such as a pet mold

robot which imitated the body mechanism of the animal of quadrapedalism and its actuation like a dog or a cat or Homo sapiens, and an ape, and actuation, or "a human mold", or the stable walk control of those progress, and the expectation for utilization has also been growing. although these leg formula mobile robot compares with a crawler type robot, it is unstable and attitude control and walk control become difficult -- rise and fall of a stairway and an obstruction -- getting over -- etc. -- it excels in the point that flexible walk / transit actuation is realizable.

[0005] Like an arm type robot, the robot of a deferment type which is implanted and used for a certain specific location works only in fixed and local workspaces, such as assembly, a sorting activity, etc. of components. On the other hand, the robot of workspace of a portable type is un-restrictive, and he can move free in a predetermined path or non-path top, and the human activity of predetermined or arbitration can be executed by proxy, or he can offer the various services which replace Homo sapiens, a dog, or other life objects.

[0006] As one of the applications of a leg formula mobile robot, vicarious execution of various kinds of difficulty activities in an industrial activity, a production activity, etc. is mentioned. For example, it is vicarious execution of the maintenance in a nuclear power plant, a thermal power station plant, and a petrochemical plant, conveyance and assembly operation of the components in a plant, cleaning in a skyscraper, and the risk activity and difficulty activity like the rescue in a fire site and others etc.

[0007] Moreover, the application of "symbiosis" or "entertainment" of a life adhesion mold, i.e., human being, is mentioned rather than above-mentioned activity exchange as other applications of a leg formula mobile robot. This kind of robot emulates the rich feeling expression using the mechanisms of operation and the limbs of a leg formula ambulatory exercise with comparatively high intelligence, such as Homo sapiens or a dog (pet). Moreover, it is required it not only performs faithfully the pattern of operation inputted beforehand, but that it should realize the lively response expression in response to a partner's language and attitudes ("it strikes"). ["it praises" or "he scolding",]

[0008] The conventional toy machine has the fixed relation between user actuation and response actuation, and cannot change actuation of a toy according to liking of a user. Consequently, a user becomes ****************** soon about the toy which repeats only the same actuation.

[0009] On the other hand, the intellectual robot has the model resulting from actuation, and realizes autonomous thinking and motion control by changing a model based on input, such as voice from the outside, and an image, a tactile sense, and opting for actuation. When a robot prepares a feeling model and an instinct model, the autonomous action according to a robot's own feeling and instinct can be expressed. Moreover, when a robot equips a picture input device and voice-input/output equipment and performs image recognition processing and speech recognition processing, it also becomes possible to realize realistic communication with human being on more advanced intellectual level.

[0010] moreover, it answers having detected the stimulus from the outside, such as user actuation, and this model is changed -- namely

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

By giving the "study effectiveness", or it does not get bored for a user, the pattern of operation which was adapted for liking can be offered.

[0011] The leg formula mobile robot of these days has high information processing capacity, and can regard as a kind of computing system. Therefore, the altitude and a series of complicated operating sequences which are constituted by the combination of the pattern of operation realized on a robot or two or more fundamental patterns of operation are built according to the same activity as computer programming.

[0012] Moreover, a robot's diffusion rate will increase increasingly from now on, and permeating deeply not only the industrial world but ordinary homes and everyday life will be expected. About the product which pursues entertainment nature, it is especially expected [that a consuming public layer without the advanced knowledge about a computer or computer programming purchases and uses a robot in many cases, and]. It is thought desirable to offer the tool which supports the thing for creating and editing a robot's operating sequence comparatively easily and efficiently by interactive processing also for such a general user, i.e., an authoring system.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] The purpose of this invention is to offer the outstanding authoring system and outstanding approach of creating a series of command/data which describe a robot's predetermined pattern of operation.

[0014] The further purpose of this invention is to offer the outstanding authoring system and outstanding approach of creating a pattern of operation using the set of the

components which specify a robot's operating state.
[0015] The further purpose of this invention is to offer the outstanding authoring system and outstanding approach of arranging each part article on a computer display, and creating a pattern of operation.

[Translation done.]

* NOTICES *
JPO and INPIT are not responsible for any
damages caused by the use of this translation.
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] This invention is made in consideration of the above-mentioned technical problem. The 1st side face The editorial department which is the authoring system or approach for creating and editing a robot's control program of operation, and offers the workplace for creating and editing this robot's scenario of operation to a user, or an edit step, It is the authoring system or approach characterized by providing the transducer which changes creation and the edited scenario into the program code which can be interpreted on said robot through said editorial department, or a conversion step.
[0017] You may make it said editorial department or an edit step provide a user with the components group which components-izes the operating state used frequently, a manipulation routine, or its chicken type, and prepares it, and the actuation screen which selects each part article of said components group by directions actuation through a coordinate designating device, and is arranged.
[0018] Or you may make it said editorial department or an edit step provide a user with the environment where the script of a predetermined programming language format is created and edited.
[0019] Moreover, you may make it said transducer or transducer change the scenario of operation described in the scenario of operation and/or script format which were expressed by arrangement of components into the mnemonic code which can be interpreted in said robot. An example of the mnemonic code said here is RCODE. The after-mentioned is yielded about the detail of RCODE.
[0020] Moreover, you may have further the means of communications or the communication link step for transmitting the program code changed in said transducer or the conversion step to said robot. At means of communications or a communication link step, short-distance wireless data transmission, such as bluetooth and .11B, is applicable, for example.
[0021] The means of communications or the communication link step said here takes out

at a time one step of program codes changed in said transducer or the conversion step, and you may make it transmit them to said robot. On the other hand, in a robot side, the program code received using the interpreter can be interpreted serially, and can be performed and/or debugged.

[0022] Moreover, in order to maintain the security of commo data, said means of communications or a communication link step enciphers the program code changed in said transducer or the conversion step, and you may make it transmit it to said robot.

[0023] Moreover, you may have the interpretation section which interprets the program code received through said means of communications or a communication link step or an interpretation step, and the drive control section which drives said robot according to the interpretation result by said interpretation section or the interpretation step or a drive control step on said robot.

[0024] In such a case, the received program code is interpreted per 1 step, and said interpretation section or an interpretation step may be made to perform it.

[0025] Generally a leg formula robot has the actuation and the posture which can be transited directly from a current posture, and the posture whose **** transition is attained via a certain actuation and posture. The robot concerning this invention may have beforehand the posture transition limit information about the posture whose **** transition is attained via the actuation and the posture in the actuation which can be transited directly from a current posture, or a posture list.

[0026] In such a case, said drive control section or a drive control step can change the contents of directions of a program code into the format in which posture transition is possible based on this posture transition limit information.

[0027] Moreover, this posture transition limit information can be held in the directed graph format which consists of arcs of operation which connect between the node which shows the posture which said robot can take, and two nodes which can be changed. Therefore, said drive control section or a drive control step can change the contents of directions of a program code into the format in which posture transition is possible easily by searching for this directed graph.

[0028] Moreover, the 2nd side face of this invention is the storage which stored physically the computer software described to perform authoring processing for created and editing a robot's control program of operation on computer system in the computer-readable format. The edit step which said computer software provides with the workplace for creating and editing this robot's scenario of operation to a user, It is the storage characterized by providing the conversion step which changes creation and the edited scenario into the program code which can be interpreted on said robot through said edit step.

[0029] The storage concerning the 2nd side face of this invention is a medium which offers computer software physically in a computer-readable format to the general purpose computer system which can perform various program codes, for example. Attachment and detachment of CD (Compact Disc), FD (Floppy Disc), MO (Magneto-Opticaldisc), etc., etc. are free for such a medium, and it is a storage of portability. Or it is also technically possible to provide specific computer system with computer software in a computer-readable format via transmission media, such as a network (for a network not to ask distinction of wireless and a cable), etc.

[0030] Such a storage defines the collaboration-relation on the structure of the computer software and the storage for realizing the function of computer software predetermined in a computer system top, or a function. If it puts in another way, by installing predetermined computer software in computer system through the storage concerning the 2nd side face of this invention, on computer system, a collaboration-operation is demonstrated and the same operation effectiveness as the authoring system or approach concerning the 1st side face of this invention can be acquired.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3.In the drawings, any words are not translated.

OPERATION

[Function] A user uses the GUI screen which the authoring tool which realized this invention offers, and creates and edits the scenario of a convention of a robot by mouse actuation. Or a user may use a text editor etc., and may create and edit a robot's control program of operation in a script format.

[0032] An authoring tool changes the scenario and the control program of a script format of operation which were created and edited into the mnemonic code called RCODE. At the time of debugging of a RCODE actuation control program, a RCODE program is taken out for every line, and it enciphers, and transmits to a robot side serially using radio means, such as bluetooth and .11B.

[0033] In the robot side of another side, it has an interpreter/debugger, middleware, a driver, etc. as activation and the debugging environment of the control program of operation described by RCODE etc.

[0034] An interpreter is a high-level-language program which reads at a time the program of one line described in the RCODE format, interprets it, and performs it. A debugger discovers the error in a RCODE program (bug), and is a program which supports the activity which corrects. Middleware is a processing engine which performs control of leg formula actuation of the walk and others in a robot etc., recognition processing of the input image from a camera, recognition processing of the voice input from a microphone, etc.

[0035] Middleware has registered beforehand the posture in which a robot 1 can change, and the actuation at the time of changing in the form of a directed graph, changes the action command from a high order program into posture transition information according to a directed graph, and publishes the concrete directions of operation to a driver.

[0036] Therefore, a general user does not need to care about detailed hardware information like the posture and the transition property of operation in an actual robot, as long as it edits a robot's 1 scenario of operation or a control program of operation is described in a script format on a GUI screen using an authoring tool. Moreover, since the difference between logical posture and directions of operation on RCODE command level, and the posture and the transition property of operation on hardware is absorbed in middleware level, in it being contrary to a robot's posture and transition property of operation, the object of debugging of a RCODE program does not become.

[0037] The purpose, the description, and advantage of further others of this invention will become [rather than] clear by detailed explanation based on the example and the drawing to attach of this invention mentioned later.

[0038]

[Embodiment of the Invention] Hereafter, the example of this invention is explained in detail, referring to a drawing.

[0039] A. The appearance configuration of the mobile robot 1 which performs the leg formula walk by the limbs with which operation is presented in this invention is shown in the block diagram 1 of an authoring system. This robot 1 is a mobile robot of the multi-joint mold constituted by using as a model the configuration and structure of an animal of having the limbs as illustration. User actuation was answered and the mobile robot 1 of this example can especially do an expression of operation while he has a side face of the pet mold robot designed by imitating the configuration and structure of a dog which are the example of representation of a pet, for example, coexists with human being in human being's living conditions.

[0040] A mobile robot 1 consists of, the idiosoma unit 2, a head unit 3, a tail 4, and Limbs 6A-6D, i.e., leg units.

[0041] The head unit 3 is arranged in the upper limit before abbreviation of the idiosoma unit 2 through the neck joint 7 with a roll, a pitch, and the degree of freedom of each shaft orientations (illustration) of a yaw. Moreover, the CCD (Charge Coupled Device: charge-coupled device) camera 15 equivalent to the "eye" of a dog, the microphone 16 equivalent to a "lug", the loudspeaker 17 equivalent to "opening", and the touch sensor 18 equivalent to tactile feeling are carried in the head unit 3. The sensor which constitutes a living body's senses besides these may be included.

[0042] The tail 4 is attached in the abbreviation Gokami edge of the idiosoma unit 2 free [a curve or rocking] through the tail joint 8 with the degree of freedom of a roll and a pitching axis.

[0043] The leg units 6A and 6B constitute a forefoot, and the leg units 6C and 6D constitute hind legs. Each leg units 6A-6D consist of combination of the femoral region units 9A-9D and the leg part units 10A-10D, and are attached in each corner of front and rear, right and left of idiosoma unit 2 base, respectively. The femoral region units 9A-9D are connected with each predetermined part of the idiosoma unit 2 by the hip joints 11A-11D with a roll, a pitch, and the degree of freedom of each shaft of a yaw. Moreover, it is connected by the knee joints 12A-12D with the degree of freedom of a roll and a pitching axis between the femoral region units 9A-9D and the leg part units 10A-10D.

[0044] the mobile robot 1 constituted like illustration making the head unit 3 shake vertically and horizontally by driving each joint actuator by the command from a control section mentioned later, or making a tail 4 wag **** -- each -- a foot -- Units 6A-6D are made to drive cooperatively, and actuation of a walk, transit, etc. can be realized.

[0045] In addition, in fact, a mobile robot's 1 joint degree of freedom is arranged for every shaft, and is offered by the rotation drive of a joint actuator (not shown). Moreover, the number of the joint degree of freedom which a mobile robot 1 has is arbitrary, and does not limit the summary of this invention.

[0046] The block diagram of the electrical and electric equipment and control network of a mobile robot 1 is typically shown in drawing 2 . As shown in this drawing, a mobile robot 1 consists of the control section 20 which performs generalization-control of the whole actuation, and other data processing, the I/O section 40, a mechanical component 50, and a power supply section 60. Hereafter, each part is explained.

[0047] The I/O section 40 contains various kinds of sensors equivalent to the senses, such as CCD camera 15 which is equivalent to a mobile robot's 1 eyes as the input section, and the microphone 16 equivalent to a lug, the touch sensor 18 equivalent to tactile feeling. Moreover, the loudspeaker 17 equivalent to opening etc. is equipped as the output section. These output section can express the user feedback from a robot 1 in the form of [other than a machine movement pattern with a foot etc.].

[0048] A mobile robot 1 can recognize the objective configuration and the color of arbitration which exist on workspace by a camera 15 being included. Moreover, the mobile robot 1 may have further the receiving set which receives dispatch waves other than a vision means like a camera, such as infrared radiation, an acoustic wave, a supersonic wave, and an electric wave. In this case, based on the sensor output which detects each carrier wave, the location and sense from the source of dispatch are measurable.

[0049] A mechanical component 50 is functional block which realizes machine movement of a mobile robot 1 according to the predetermined movement pattern which a control section 20 orders it, and consists of drive units prepared for every shafts, such as a roll in each joint, such as the neck joint 7, the tail joint 8, hip joints 11A-11D, and knee joints 12A-12D, a pitch, and a yaw. A mobile robot 1 has the joint degree of freedom of n pieces, therefore a mechanical component 50 is constituted from the example of illustration by n drive units. Each drive unit consists of combination of the driver 53 which controls the rotation location and rotational speed of a motor 51 accommodative based on the output of the motor 51 which performs rotation actuation of the circumference of a predetermined shaft, the encoder 52 which detects the rotation location of a motor 51, and an encoder 52.

[0050] A power supply section 60 is a functional module which supplies electric power to each electrical circuit in literal [the] and a mobile robot 1 etc. The mobile robot 1 concerning this example is the autonomous drive type which used the dc-battery, and a power supply section 60 consists of a charge dc-battery 61 and a charge-and-discharge control section 62 which manages the charge-and-discharge condition of the charge dc-battery 61.

[0051] The charge dc-battery 61 consists of gestalten of the "battery pack" which package-ized two or more nickel cadmium battery cels to the cartridge-type.

[0052] Moreover, the charge-and-discharge control section 62 grasps the remaining capacity of a dc-battery 61 by measuring the terminal voltage of a dc-battery 61, charge/strength of discharge current, the ambient temperature of a dc-battery 61, etc., and determines an initiation stage, a termination stage, etc. of charge. The initiation and the termination stage of charge which the charge-and-discharge control section 62 determines are notified to a control section 20, and serve as a trigger for a mobile robot 1 to start and

end charge operation. The after-mentioned is yielded about the detail of charge operation.
[0053] A control section 20 is equivalent to "brains", for example, is carried in a mobile robot's 1 head unit 3 or idiosoma unit 2.
[0054] The configuration of a control section 20 is further illustrated in the detail at drawing 3 . As shown in this drawing, the control section 20 has the composition that the bus connection of CPU (Central Processing Unit)21 as a Maine controller was carried out to each circuit component, which are memory and others, or a peripheral device. The address (a memory address or I/O Address) of a proper is assigned at each to each equipment on a bus 27, and CPU21 can communicate with the specific equipment on a bus 28 by addressing.
[0055] RAM (Random Access Memory)22 is the memory which consisted of volatile memory, such as DRAM (Dynamic RAM), and which can be written in, loads the program code which CPU21 performs, or is used for temporary preservation of activity data.
[0056] ROM (Read Only Memory)23 is a read-only memory which stores a program and data everlastingly. The self-test test program performed to a mobile robot's 1 power up, the control program which specifies actuation of a mobile robot 1 are mentioned to the program code stored in ROM23.
[0057] The "sensor input-process program" which processes sensor inputs, such as a camera 15 and a microphone 16, the "action instruction program" which generates a mobile robot's 1 action, i.e., a movement pattern, based on a sensor input and a predetermined model of operation, and the "drive control program" etc. which controls the drive of each motor, the voice output of a loudspeaker 17, etc. according to the generated movement pattern are contained in a robot's 1 control program. High actuation of entertainment nature, such as a "hand", a "rain check", "stability", and utterance of the cry of animals, such as "one one", may be included in the movement pattern generated in addition to usual locomotion and usual transit movement.
[0058] Moreover, creation and various kinds of edited operating-sequence programs are included, using an authoring tool as a control program of others of a robot 1. Although an authoring tool is started under a software execution environment predetermined in an external computer system top, it is explained in detail to the program created and edited by the authoring tool list on this tool, therefore the back.
[0059] Like EEPROM (Electrically Erasable and Programmable ROM), nonvolatile memory 24 consists of memory devices in which elimination re-writing is possible electrically, and it is used in order to hold the data which should be updated serially in un-volatilizing. The model which specifies a mobile robot's 1 behavior pattern is mentioned to the data which should be updated serially.
[0060] An interface 25 is equipment for interconnecting with the device besides a control section 20, and making the data exchange possible. An interface 25 performs data I/O between a camera 15, a microphone 16, and a loudspeaker 17. Moreover, an interface 25 is each driver 53-1 in a mechanical component 50. -- I/O of data or a command is performed in between. Moreover, an interface 25 can also perform transfer of charge initiation and a charge terminate signal among power supply sections 60.
[0061] An interface 25 Serial interface, such as RS(Recommended Standard)-232C, Parallel interfaces, such as IEEE (Institute of Electrical and electronics Engineers)1284, A USB (Universal Serial Bus) interface, An i-Link (IEEE1394) interface, a SCSI (Small

Computer System Interface) interface, etc. It has a general interface for peripheral-device connection of a computer, and may be made to perform program and migration of data between the external instruments by which local connection was made.

[0062] Moreover, it has an infrared-ray-communication (IrDA) interface as one of the interfaces 25, and may be made to perform an external instrument and radio. As for the transceiver section for infrared ray communication, it is desirable from a viewpoint of receiving sensibility to be installed in the point of mobile-robot 1 bodies, such as the head unit 2 and a tail 3.

[0063] Furthermore, a control section 20 can perform an external host computer 100 and data communication via contiguity radio as shown in "bluetooth" or ".11B" or LAN (Local Area Network: (trademark), for example, Ethernet), or the Internet including radio interface 26 Network Interface Card (NIC) 27.

[0064] The purpose of the data communication between such a robot and a host computer is operating actuation of a mobile robot 1 by remote control using a remote computer resource. Moreover, other purposes of this data communication are to supply data and program of the robots 1, such as a model of operation and other program codes, required for motion control to a mobile robot 1 via a network. Moreover, other purposes of this data communication are debugging processings of the real time according using an authoring tool to collaboration-actuation with download of the program for creation and the robot motion control which edited, the host computer 100 of such a program for motion control, and a robot 1 on a host computer 100. However, about an authoring tool or debugging processing, it mentions later behind.

[0065] Moreover, a control section 20 may be equipped with the keyboard 29 which consists of a ten key and/or an alphabet key. A keyboard 29 is used for the direct command input in a work site, and also it is used for the input of owner authentication information, such as a password.

[0066] The robot 1 concerning this example can perform autonomous actuation, when a control section 20 performs a predetermined control program. Moreover, while having an input unit equivalent to the senses of human beings, such as an image input (namely, camera 15), voice input (namely, microphone 16), and a touch sensor 18, or an animal, it has the intelligence which performs reasonable or emotional actuation which answered these external inputs.

[0067] The mobile robot 1 constituted as shown in drawing 1 - drawing 3 has the following descriptions. Namely, [0068] (1) When changing from a certain posture to other postures is directed, don't change between each posture directly but change via an in-between posture without the unreasonableness prepared beforehand.

(2) A notice can be received when the posture of arbitration is reached by posture transition.

(3) In each unit unit, such as a head, a foot, and the tail section, a posture can be managed independently and can carry out attitude control. Namely, apart from the whole robot's 1 posture, a posture is manageable for every unit.

(4) A parameter to show the detail of actuation of an instruction of operation can be passed.

[0069] As shown in drawing 3 , the mobile robot 1 concerning this example interconnects with the external host computer 100 via the network. Or the means of communications of radio (for example, bluetooth and .11B short distance wireless data transmission) or others may connect in the host computer 100.

[0070] On a host computer 100, a predetermined software execution environment is built, under this environment, an authoring tool can be started and a robot's 1 operating sequence can be created comparatively easily and efficiently by interactive processing. About the detail of an authoring tool, it mentions later.

[0071] In drawing 4 , the example of a hardware configuration of a host computer 100 is illustrated typically. Hereafter, each part in a computer 100 is explained.

[0072] CPU (Central Processing Unit)101 which is the Maine controller of a system 100 performs various kinds of applications under control of an operating system OS. Although OS offers the GUI (Graphical User Interface) environment more preferably, it is good at Windows 98 [UNIX (trademark) or] of U.S. Microsoft/NT, for example.

[0073] CPU101 interconnects with other equipments (after-mentioned) by bus 107 as illustration. The memory address or I/O Address of a proper is given to each device on a bus 107, respectively, and device access is possible for CPU101 by these addresses. An example of a bus 107 is a PCI (Peripheral Component Interconnect) bus.

[0074] Memory 102 is storage used since the program code performed in a processor 101 is stored or the activity data under activation are stored temporarily. Please understand the memory 102 shown in this drawing to be a thing containing both un-volatilizing and volatilization memory.

[0075] The display controller 103 is an exclusive controller for actually processing the drawing instruction which a processor 101 publishes, for example, supports a bit map drawing function equivalent to SVGA (Super Video Graphic Array) or XGA (eXtended Graphic Array). Once it is written in a frame buffer (not shown), the screen output of the drawing data processed in the display controller 103 is carried out at a display 111. Indicating equipments 111 are for example, a CRT (Cathode Ray Tube) display, a liquid crystal display display (Liquid Crystal Display), etc.

[0076] The input device interface 104 is equipment for connecting user input devices, such as a keyboard 112 and a mouse 113, to a system 100. The input device interface 104 answers the coordinate directions input through the key input or mouse 113 by the keyboard 112, and generates interruption to a processor 101.

[0077] According to predetermined communications protocols, such as Ethernet, it can connect with networks, such as LAN (Local Area Network), or the network interface 105 can connect a system 100 to short-distance wireless data transmission like bluetooth or .11B. Generally, the network interface 105 is offered with the gestalt of a LAN adapter card, and the PCI bus slot on a mother board (not shown) equips with it, and it is used.

[0078] In the example shown in drawing 3 , although the host computer 100 interconnects with the robot 1 via wireless data transmission or a network, of course, both may be connected by other means of communications.

[0079] Moreover, on the network, two or more host computers (not shown) are connected in the transparent condition, and the distributed computing environment is built. Distribution of a software program, data contents, etc. is performed on a network. For example, the authoring tool concerning this example, the operating-sequence program for robots created and edited by this authoring tool can be distributed via a network. Moreover, distribution service of such a program/data may be offered the charge or for nothing.

[0080] The external instrument interface 106 is equipment for connecting external

devices, such as a hard disk drive (HDD) 114 and the media drive 115, to a system 100. The external instrument interface 106 is based on interface specification, such as IDE (Integrated Drive Electronics) and SCSI (Small Computer System Interface).
[0081] HDD114 is the external storage which carried the magnetic disk as storage support fixed (common knowledge), and excels other external storage in points, such as memory capacity and a data transfer rate. It calls it "install" to the system of a program to place on HDD116 in the condition that a software program can be performed. Usually, the program code of the operating system which a processor 511 should perform, an application program, a device driver, etc. are stored in HDD114 in un-volatilizing. For example, creation and the edited operating-sequence program for robots are installable on HDD114 using the authoring tool concerning this example, and this authoring tool.
[0082] Moreover, the media drive 115 is equipment for loading with portable mold media, such as CD (Compact Disc), and MO (Magneto-Optical disc), DVD (Digital Versatile Disc), and accessing a data-logging side. Portable mold media are used in order to mainly move backing up a software program, a data file, etc. as data of a computer-readable format, and these between systems (sale, circulation, and distribution are included). For example, these portable mold media can be used, and the authoring tool concerning this example, the operating-sequence program for robots created using this authoring tool can be physically circulated and distributed between devices.
[0083] In addition, an example of the host computer 100 as shown in drawing 4 is the compatible machine or succeeding machine of personal computer PC/AT (Personal Computer/Advanced Technology) of U.S. IBM. Of course, it is also possible to apply the computing system equipped with other architecture to the host computer 100 concerning this example.
[0084] In this example, creation and edit of the control program of operation which consists of a series of command/data which describe a robot's 1 predetermined pattern of operation are done using the authoring tool started on the host computer 100. Moreover, creation and the edited control program of operation are transmitted to a robot 1 side using radio means, such as bluetooth and .11B, using this authoring tool, and collaboration-actuation with a host computer 100 and a robot 1 performs debugging processing. That is, the authoring system for a robot's 1 control program of operation is built by induction-association of a host computer 100 and a robot 1.
[0085] In drawing 5 , the whole authoring system configuration is illustrated typically.
[0086] In a host computer 100 side, a user can use the GUI (Graphical User Interface) screen which an authoring tool offers, and can create and edit the scenario of a convention of a robot 1 by mouse actuation. About the actuation on this screen, it mentions later in the GUI screen list for scenario creation. Or a user can use a text editor etc., and can create and edit a robot's 1 control program of operation in a script format (for example, high level language formats, such as C).
[0087] An authoring tool changes the scenario which the user created and edited on the GUI screen, and the control program of the script format created and edited on the text editor of operation into a mnemonic code like the assembler called "RCODE".
[0088] RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, and since it also has fundamental control structures, such as IF and GO, it can be used also as a minimum level script language for robot controls. However, about the detail of RCODE, it explains in detail behind.

[0089] The RCODE actuation control program created and edited on the host computer 100 is movable to a robot 1 side using media, such as a memory stick. Moreover, at the time of debugging of a RCODE actuation control program, a RCODE program is taken out for every line, and it enciphers, and transmits to a robot 1 side serially using radio means, such as bluetooth and .11B.
[0090] On the other hand, in the robot 1 side, it has an interpreter/debugger, middleware, the driver, and the operating system (OS) as activation and the debugging environment of the control program of operation described by RCODE etc.
[0091] An interpreter is a high-level-language program which reads at a time the program of one line described in the RCODE format, interprets it, and performs it. However, in the time of debugging etc., when a RCODE program is transmitted in the format enciphered from the host computer 100 side, once an interpreter decrypts this, it performs interpretation and activation.
[0092] A debugger discovers the error in a RCODE program (bug), and is a program which supports the activity which corrects. That is, activation can be stopped in the line which specified the program, or the memory at that time and the contents of the variable can be referred to.
[0093] Middleware is a processing engine which performs control of leg formula actuation of the walk and others in a robot 1 etc., recognition processing of the input image from a camera 15, recognition processing of the voice input from a microphone 16, etc.
[0094] A driver is a program code for operating hardware of each joint actuator or others.
[0095] In this example, the driver is mounted in the middleware list by the object-oriented program. The software based on object-oriented is fundamentally treated in the module unit of the "object" which made the processing procedure over data and its data unify. Moreover, if needed, two or more objects are created or one software is completed by combining. Generally, according to object-oriented, it is thought that the efficiency of development and maintenance of software is increased.
[0096] An operating system (OS) performs control about management of the data communication between these objects, and other program executions. OS is also mounted by the object-oriented program.
[0097] B. The GUI screen which the authoring tool concerning this example offers by the creation / edit following ** of the program for robots of operation using an authoring tool for a user to create and edit a robot's 1 scenario of operation is explained. The configuration of this GUI screen is illustrated in drawing 6 .
[0098] As shown in this drawing, this GUI screen includes the edit window which performs an editing task by making click by the mouse, and drag-and-drop actuation into the keynote, and the components window which components-izes operations sequence frequently used in an editing task, and offers it.
[0099] As for the edit window, the title bar, the menu bar, the tool bar, and the edit field are arranged in order [upper case / maximum].
[0100] A menu bar is the field where the user arranged the list of selectable processing in the longitudinal direction by click actuation, and each menu item "a file (F)", "edit (E)", "a display (V)", "insertion (I)", and "a group (G)" is prepared in this example. If a certain menu item is chosen, the pull down menu which corresponds further will appear.
[0101] In drawing 7 , the pull down menu about a menu item "a file" is illustrated. As

shown in this drawing, in this pull down menu, each sub menu "new creation", "it opening", "overwriting", "naming and saving an identifier", "creation of a memory stick image", "a transfer of a memory stick image", and "termination" is arranged.

[0102] Selection of "new creation" creates newly the scenario (henceforth a "project") which describes a series of actuation of a robot. When the project under edit already exists on an edit window, the dialog to which the preservation is urged appears. The edit window shown in drawing 6 is an initial screen when creating a new project, and the Maine group who consists only of a "START" box and "END" box which corresponds to the head and the tail end of a project, respectively is displayed on the edit field.

[0103] Selection of a sub menu "it opens" opens the project saved as a file on an edit window. When the project under edit already exists on an edit window, the dialog to which the preservation is urged is displayed. In this example, a project shall be treated as a file with extension "apk".

[0104] Selection of a sub menu "overwriting" saves the project under edit to the file of a same name. The dialog which asks a file name shall not appear at this time.

[0105] Selection of a sub menu "name and save an identifier" displays the dialog which asks a file name. On this dialog, a user can specify the directory name and file name of a preservation place, and a project is saved by the file name to the specified directory.

[0106] Selection of a sub menu "creation of a memory stick image" creates the image file for writing in a memory stick (or storage media equivalent to this). At this example, an image file shall be saved by extension called "aim" to the same directory as the file of the original apk format.

[0107] If a sub menu "a transfer of a memory stick image" is chosen, it will generate to the full path which had the image file on a memory stick specified.

[0108] Selection of a sub menu "termination" ends the authoring tool under activation.

[0109] Moreover, to drawing 8 , the pull down menu about a menu item "a group" is illustrated. As shown in this drawing, in this pull down menu, "new group creation", "deletion of this group", "a group list", a "group layered structure", "all groups being opened", and each sub menu of "closing all groups" are arranged.

[0110] Selection of a sub menu "new group creation" creates a new group.

[0111] Selection of a sub menu "deletion of this group" deletes the group by whom it is indicated by current. However, even if it deletes a group, the group box which is calling the group is not deleted.

[0112] Selection of a sub menu "a group list" displays a list of a group. The group can be opened in double-clicking a group's displayed identifier.

[0113] Selection of a sub menu "a group layered structure" displays a group's layered structure (call structure). The group can be opened in double-clicking a group's displayed identifier.

[0114] Selection of a sub menu "all groups are opened" opens all groups' window.

[0115] Selection of a sub menu "all groups are closed" closes all groups' window (however, the Maine group removes).

[0116] Moreover, in the example shown in drawing 6 , each tool carbon button the "normal mode", "connection mode", "a box of operation", a "branching box", and a "group box" is arranged in the tool bar.

[0117] If a carbon button "the normal mode" is chosen, mouse actuation will change to the normal mode. For example, the drag actuation to a box functions as migration of a box.

[0118] If a carbon button "connection mode" is chosen, mouse actuation will change in connection mode. That is, the drag actuation to a box functions as connection between boxes. What is necessary is just to perform drag actuation from a connecting agency to the box of a connection place. Moreover, when it is connection from a branching box, the dialog which specifies branch condition is displayed.

[0119] Selection of a carbon button "a box of operation" inserts a box of operation (Action) in a downward edit field.

[0120] Selection of a carbon button "a branching box" inserts a branching (Branch) box in a downward edit field.

[0121] Selection of a carbon button "a group box" inserts a group box in a downward edit field.

[0122] A user can specify processing with the combination of mouse actuation of a click, a drag, etc., mouse actuation, and a key input on the edit field shown in drawing 6 .

[0123] For example, a box can be chosen by the click of the left carbon button of a mouse.

[0124] Moreover, when the left carbon button of a mouse is double-clicked, a property is displayed and/or a group is opened.

[0125] Moreover, when the left carbon button of a mouse is pushed and dragged under the normal mode and a mouse cursor suits on a box, it moves in a box, and a range selection is performed when a mouse cursor is outside a box. Moreover, boxes are connected when the left carbon button of a mouse is pushed and dragged under connection mode, and a mouse cursor is on a box.

[0126] Moreover, if the combination of the depression/drag of the left carbon button of a mouse is performed the bottom of the [Ctrl] key press under the normal mode, when a mouse cursor suits on a box, a box is connected, and range assignment will be performed when a mouse cursor is outside a box. Moreover, if the combination of the depression/drag of the left carbon button of a mouse is performed the bottom of the [Ctrl] key press under connection mode, when a mouse cursor is on a box, it will move in a box.

[0127] Moreover, an edit window will be resized if the combination of the left carbon button depression / drag of a mouse is performed the bottom of the [Shift] key press.

[0128] Moreover, a click of the right carbon button of a mouse displays a context menu.

[0129] In addition, drag actuation of a mouse is canceled by the depression of the [Esc] key.

[0130] Subsequently, the basic operation procedure on the GUI screen which an authoring tool as shown in drawing 6 offers is explained.

[0131] First, a new project is created by choosing the sub menu [new creation] of a menu [a file]. As the screen of a new project is shown in drawing 6 , the Maine group who consists only of a "START" box and "END" box which corresponds to the head and the tail end of a project, respectively is displayed on the edit field.

[0132] Subsequently, the box which corresponds on an edit window can be arranged by choosing a menu [insertion] or the carbon button in a tool bar [A], [B], and [G]. The box once arranged on an edit window can move an installation location by mouse actuation.

[0133] In drawing 9 , signs that a box [A] of operation and one branching box [B] are arranged each on the edit window are illustrated. Although "Box002" is given to the

default identifier "Box001" list, a user can change this into each box later.
[0134] Subsequently, mouse actuation is switched to connection mode by choosing the sub menu [mouse:connection mode] of a menu [edit], or choosing the [connection mode] carbon button in a tool bar. And the boxes in the context on scenario advance are connected by mouse actuation.
[0135] In drawing 10 , signs that the box [A] of operation and the branching box [B] were connected to the [START] box, the box [A] of operation, and the list are illustrated on the edit window shown in drawing 9 .
[0136] Since the dialog (not shown) for specifying the branching surface opens automatically about the connection from a branching box [B] in case boxes are connected, the user should just specify conditions according to the directions in a dialog. However, connection is not made, unless criteria specification of a branching box shall not be made to deferment rather than the time of connection with the next box and it specifies conditions completely.
[0137] Moreover, the dialog for specifying the detail of processing actuation of the box concerned opens a user in double-clicking the left carbon button of a mouse on a certain box. Moreover, when a group box is double-clicked, the group opens as another edit window (after-mentioned).
[0138] Processing actuation of each box is described by this example by mnemonic code like an assembler called RCODE. Although RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, about the detail of RCODE, it explains in detail behind.
[0139] In drawing 11 , the dialog for specifying the detail of a box of operation is illustrated. This dialog can be called by double-clicking a desired box [A] of operation on an edit window as shown in drawing 9 or drawing 10 .
[0140] A user can specify the identifier of a box of operation by inputting a character string into the identifier field on a dialog as shown in drawing 11 . Moreover, the comment about this box of operation can be filled in on a comment field.
[0141] Whether the direct input of the command name and command parameter of RCODE is carried out into combo boxes, such as Action, Part, Sound, and Volume#, furthermore, or the right end of this box By choosing a desired command or a desired command parameter from the list box (not shown) which pushes a carbon button and appears, one line, i.e., the RCODE command for one step, can be set up.
[0142] If the RCODE command for one step is set up using combo boxes, such as these Action(s), Part, Sound, and Volume#, and a carbon button [additional (A)] is clicked further, sequential registration will be carried out on a command list.
[0143] Moreover, it is on this command list, and after choosing a predetermined line, by clicking [modification (M)] carbon button, the line concerned serves as a candidate for modification, and those contents of a setting are displayed on each combo box, such as Action, Part, Sound, and Volume#. Moreover, it is on this command list, and after choosing a predetermined line, the line concerned can be deleted from a command list by clicking [deletion (D)] carbon button.
[0144] And if assignment of the detail of a box of operation is completed on this dialog, while the contents of assignment which minded the screen by clicking the closed [(C)] carbon button will be registered as a manipulation routine described by RCODE, a dialog closes.

[0145] Moreover, to drawing 12 , the dialog for specifying the detail of a branching box is illustrated. This dialog can be called by double-clicking a desired branching box [B] on an edit window as shown in drawing 9 or drawing 10 .

[0146] A user can specify the identifier of a branching box by inputting a character string into the identifier field on a dialog as shown in drawing 12 . In the example of illustration, "mode branching" is filled in as an identifier of the branching box concerned. Moreover, the comment about this branching box can be filled in on a comment field.

[0147] Whether the direct input of the command name and command parameter of RCODE is carried out into combo boxes, such as Type and Variable, furthermore, or the right end of this box By choosing a desired command or a desired command parameter from the list box (not shown) which pushes a carbon button and appears, the RCODE command which describes the conditional judgment of the branching box concerned can be set up.

[0148] If the RCODE command of ********** is set up using combo boxes, such as these Type(s) and Variable, and a carbon button [additional (A)] is clicked further, sequential registration will be carried out on a command list.

[0149] Moreover, it is on this command list, and after choosing a predetermined line, by clicking [modification (M)] carbon button, the line concerned serves as a candidate for modification, and those contents of a setting are displayed on each combo box, such as Type and Variable. Moreover, it is on this command list, and after choosing a predetermined line, the line concerned can be deleted from a command list by clicking [deletion (D)] carbon button.

[0150] And if assignment of the detail of a branching box is completed on this dialog, while the contents of assignment will be registered by clicking the closed [(C]) carbon button as a manipulation routine described by RCODE, a dialog closes.

[0151] In addition, there is not necessarily no need that a user sets up the detail of each box himself using a dialog as shown in drawing 11 or drawing 12 . Since a box of operation, a branching box, etc. which completed detailed assignment beforehand components-ize in the components window on the GUI screen which an authoring tool prepares (for example, refer to drawing 6) and are prepared for it, a user can use the components concerned as it is as a part of program by drags and drops a desired components box from a components window to an edit window.

[0152] Moreover, on the GUI screen (refer to drawing 6) which an authoring tool offers, if the right carbon button of a mouse is clicked on a certain box, "Contextual Menu" (not shown) will appear. [Deletion of a box] and [deletion of connection] can be performed from this Contextual Menu.

[0153] However, on a GUI screen, the specific line which connects between boxes cannot be specified, and the deletion cannot be directed directly, but Contextual Menu must be used for deletion of a line. In the case of a branching box, the dialog (not shown) containing the list box for choosing which connection is deleted appears. A user should just click [deletion] carbon button from which connection to delete out of this list was chosen.

[0154] A user can create the program which described, a scenario, i.e., the operations sequence, to which a robot 1 performs a soccer game, using a GUI screen of an authoring tool which was mentioned above. Such a scenario is described by the box configuration as shown in drawing 13 .

[0155] In the example shown in drawing 13 , after performing a robot's 1 initial setting from the contents specified in the box [initialization] of operation, according to a comparison result with the conditions specified with the branching box [mode branching], a robot 1 performs one mode of operation among [fall recovery], the [search mode], and a [trace mode].

[0156] Here, each of [fall recovery], the [search mode], and a [trace mode] is described as a group box. As mentioned above, when a group box is double-clicked on an edit window, the group opens as another edit window. In drawing 14 , the edit window of the search mode which double-clicks the [search mode] and appears is illustrated. Moreover, to drawing 15 , the edit window of a trace mode which double-clicks a [trace mode] and appears is illustrated.

[0157] By this example, being described by mnemonic code like an assembler called RCODE already stated processing actuation of a robot 1. Here, the detail of RCODE is explained.

[0158] RCODE is the programming language upon which it was decided in order to control a robot 1 by the easy command, and since it also has fundamental control structures, such as IF and GO, it can be used also as a minimum level script language for robot controls.

[0159] In RCODE, the word of all capital letters is reserved for the constant identifier etc., and cannot be used for the variable name which a user prepares.

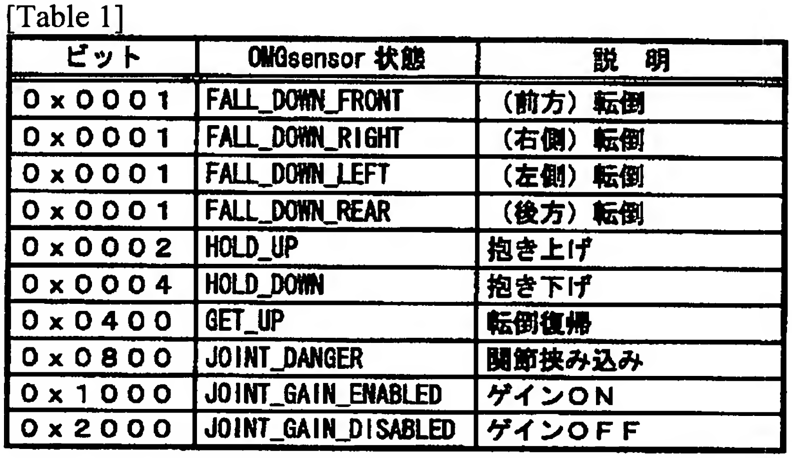
[0160] RCODE is equipped with the operator (operator) for specifying the action which should be performed for control, a jump, branching, substitution, the synchronization, the function, etc. Below, the list of RCODE operators is shown.

[0161]
[Equation 1]
<<control>>
EDIT It shifts to the edit mode (transfer of the code which lets serial Rhine pass).
RUN Activation is started.
END The /activation which ends the edit mode is ended.
LOAD: <filename> SAVEwhich reads the code on a memory stick on memory :
<filename> <<jump>> which writes in the code on memory on a memory stick
: <label> Label definition. <label> is one or more numeric values (0 expresses the following line).
GO: <lebel> It jumps on a label.
<<branching>>
IF: <op>:<var1>:<const>|<var2>: [<then_label> : <else_label>] A variable 1 <var1>, a constant <const>, or a variable 2 <var2> is compared, and it branches.
<op> " -- = -- " -- " -- < -- > -- " -- " -- < -- " -- " -- < -- = -- " -- " -- > -- " -- " -- > -- = -- " -- six -- a kind -- it is .
SWITCH: <var> The value of a variable is set up as a context value.
CSET: <op>:<var1>|<const1>:<var2>|<const2>:<var3>| <const3> The <op> operator compares a value 1 and a value 2, and if the result is truth, a value 3 will be set up as a context value.
When a CSET instruction appears continuously, the CSET instruction after the CSET instruction which became truly serves as No Operation treatment (****** instruction).
CASE: <const>: <RCODE command> <RCODE command> will be performed if a

context value is equal to <const>.
<<substitution>>
LET: <var1>:<const>| <var2> <var1> <- <const> | <var2> Substitution
SET:<var1>:<const>| of the value to a variable <var2> <var1> <- <const> | <var2> A special function works depending on a variable. (It is not mere substitution)
GET: <var1> The contents of <var1> are displayed on a console (for debugging).
<<operation>>
ADD: <var1>:<const>| <var2> <var1><-<var1>+<const>| <var2> Addition
SUB:<var1>:<const>| <var2> <var1> <-<var1>-<const> | <var2> Subtraction
MUL:<var1>:<const>| <var2> <var1><-<var1>*<const>| <var2> Multiplication
DIV:<var1>:<const>| <var2> <var1><-<var1>/<const>| <var2> Division
MOD:<var1>:<const>| <var2> <var1><-<var1>%<const>| <var2> Remainder
AND:<var1>:<const>| <var2> <var1><-<var1>&<const>| <var2> AND
IOR:<var1>:<const>| <var2> <var1><-<var1>|<const>| <var2> OR
XOR:<var1>:<const>| <var2> <var1><-<var1>^<const>| <var2> Exclusive-OR
RND:<var1>:<from>: <to> <var1> <- range <from>- <to> Uniform-random-number value <<synchronous>>
WAIT It waits for termination (reproductive termination) of the actuation performed immediately before.
WAIT: <ms> It waits only for the ms made into the argument. <ms> = 1-30000. (Notes: Resolution 32 [ms])
WAIT:SOUND It waits for termination of sound playback (it can use also for the judgment under sound playback independently).
WAIT:LIGHT It waits for termination of photoregeneration (it can use also for the judgment under photoregeneration independently).
<<function>>
CALL: <label> <label> A subroutine is called.
* A nest is a maximum of 16 pieces. Suppose parameter delivery that a variable is substituted.
RETURN Return from a subroutine .
* Since the stack pointer did not check, it needs to be cautious of the adjustment of a nest.
[0162] The command which controls "a motion" of a robot 1 among the commands of RCODE is divided roughly into the following five basic forms. Namely, [0163] (1) POSE It is made to take a certain posture (initiation posture of a motion).
(2) MOVE A certain motion is carried out. OMNE/OMTE/OMLE2_S/OMSE Correspondence.
(3) PLAY A certain motion is reproduced.
OAllReplay/OHeadReplay/OLegsReplay/OTailReplay It corresponds.
(4) STOP Halt of operation (since the current line ends the motion which is).
(5) QUIT Emergency shut down of operation (immediately).
[0164] MOVE and PLAY can be said to be being only the difference in whether inclusion actuation is performed or actuation defined as the "*.mtn" file on a memory stick is performed, and being the same function at the point of "carrying out a certain actuation." MOVE Although it is not customizable since actuation is inclusion, fine parameter assignment etc. can be performed. On the other hand, it is substitutable, although parameter assignment cannot be performed since actuation of PLAY is playback

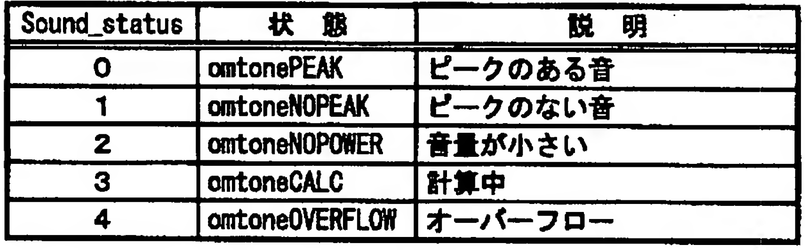
of a "*.mtn" file.
[0165] Below, the list of the RCODE commands is shown.
[0166]
[Equation 2]
SET:ROBOT:POWER: One Power-source ONSET:ROBOT:POWER: Zero Power-source OFFGET:ROBOT All variable discharge GET:ROBOT: <var> Variable Acquisition SET:ROBOT:<var>: <value> Variable Setting POSE:ROBOT: <motion_name> To the initiation pause of motion POSE:LEGS: <motion_name> To the initiation pause of motion POSE:HEAD: <motion_name> It is POSE:TAIL: to the initiation pause of motion <motion_name>. It is PLAY:ROBOT: to the initiation pause of motion <motion_name>. motion playback PLAY:LEGS: <motion_name> motion playback PLAY:HEAD: <motion_name> motion playback PLAY:TAIL: <motion_name> motion playback PLAY:SOUND:<sound_name>: <volume> Sound playback volume =:[0-100PLAY:LIGHT] <pattern_name>: <times> Photoregeneration (LED pattern)
times = 0 - 16 MOVE:ROBOT (0:Default 16:Loop) : <motion_name> OMLE inclusion actuation playbacks (fall return etc.)
MOVE:LEGS:WALK: <style1>:<dir>: <times> Walk
MOVE:LEGS:STEP:<style1>:<dir>: <times> Walk (StepWalk)
MOVE:LEGS:KICK: <style2>: <deg> A kick 12 (R-Turn) style1 = 0-10, 13 (L-Turn) style 2 = 14 (R-Kick), 15 (L-Kick) dir = [1-6] times = [0-9999] deg =-90 - 90 MOVE:HEAD:HOME head home-position MOVE:HEAD:ABS:<tilt>:<pan>:<roll>: <time> Head Absolutely Impaction efficiency MOVE:HEAD:REL:<tilt>:<pan>:<roll>: <time> head Relative-position migration MOVE:HEAD:C-TRACKING Head color tracking MOVE:HEAD:C-TRACKING: <time> Head color tracking tilt =-180 - 180 pan =-180-180 roll =-180-180 time = 0-99999 [ms] MOVE:TAIL:HOME Tail Home-position MOVE:TAIL:ABS: <tilt>:<pan>: <time> Tail absolute impaction efficiency MOVE:TAIL:SWING:<tilt>:<pan>: <time> Tail swing tilt =-90-90 pan =-90-90 time = 0-99999 [ms] whole STOP:ROBOT usual halt STOP:LEGS Limbs usual halt STOP:HEAD Head usual halt STOP:TAIL Tail usual halt STOP:SOUND Sound playback halt STOP:LIGHT Photo-regenerating halt QUIT:ROBOT Whole emergency shut down QUIT:LEGS limbs emergency shut down QUIT:HEAD Head emergency shut down QUIT:TAIL tail emergency shut down QUIT:SOUND (STOP:SOUND and equivalence) sound playback halt QUIT:LIGHT (STOP:LIGHT and equivalence) Photo-regenerating halt [0167] Moreover, RCODE defines the system variable, in order to direct actuation / control value or to acquire the condition of each part to a robot's 1 each part (read-out of a sensor output value etc.). Below, the list of the system variables of RCODE is shown.
[0168]
[Equation 3]
Power A power source 0:-1:ONHead_tilt 1st joint include-angle [**] Head_pan of a head 2nd joint include-angle [**] Head_roll of a head 3rd joint include-angle [**] Head_mouth of a head Head temporomandibular-joint include-angle [**] Tail_1 1st joint include-angle [**] Tail[of a tail]_2 2nd joint include-angle [**] Leg_RF[of a tail]_1 1st joint include-angle [**] Leg_RF[of the right nose gear]_2 2nd joint include-angle [**] Leg_RF[of the right nose gear]_3 3rd joint include-angle [**] Leg_LF[of the right

nose gear]_1 1st joint include-angle [**] Leg_LF[of the left nose gear]_2 2nd joint include-angle [**] Leg_LF[of the left nose gear]_3 3rd joint include-angle [**] Leg_RR[of the left nose gear]_1 the right hind leg 1st -- joint include-angle [**] Leg_RR_2 2nd joint include-angle [**] Leg_RR[of right hind legs]_3 3rd joint include-angle [**] Leg_LR[of right hind legs]_1 1st joint include-angle [**] Leg_LR[of left hind legs]_2 2nd joint include-angle [**] Leg_LR[of left hind legs]_3 3rd joint include-angle [**] Head_sw of left hind legs Head pressure sensor [10^-3Pa] Distance Head obstruction sensor [mm] Leg_RF_sw Right nose-gear **** sensor [On:-1 Off:0] Leg_LF_sw Left nose-gear **** sensor [On:-1 Off:0] Leg_RR_sw Right hind leg **** sensor [On:-1 Off:0] Leg_LR_sw Left hind leg **** sensor [On:-1 Off:0] Gsensor_status G sensor status 16-bit flag Gsensor_roll G sensor roll angle [**] Gsensor_pitch G sensor pitch angle [**] Gsensor_yaw G sensor yaw angle [**] Cdt_npixel Number [Pixels] Psd_status of color-sensor pixels Obstruction sensor situation Psd_range Distance Touch_head to an obstruction Head touch sensor Touch_head_time Head touch sensor depression time amount Touch_head_press Head touch sensor mean pressure [unit -- unknown --]Touch_RF right nose-gear **** sensor ON->OFF/OFF->ON detection Touch_LF Left nose-gear **** sensor ON->OFF/OFF->ON detection Touch_RR right hind leg **** sensor ON->OFF/OFF->ON detection Touch_LR Left hind leg **** sensor ON->OFF/OFF->ON detection Tone_num Tone detection Sound number Tone_level Loudness level Tone_dir Detection direction Melody_id Melody detection Melody IDMelody_num The number of tones (1-3)
Tone1_num Tone first-sound number
Tone1_level Magnitude Tone1_dir of a Tone first sound Tone1 detection direction
Tone2_num Tone second-sound number
Tone2_level Magnitude Tone2_dir of a Tone second sound Tone2 detection direction
Tone3_num Tone3 sound number
Tone3_level Tone3 loudness-level Tone3_dir Tone3 detection direction Sound_status Sound detection Status Sound_num Sound number
Sound_level Loudness level Sound_dir Detection direction Sound_busy Under 1:sound playback 0-:-sound-less Light_busy Under 1:Light playback [with no 0:Light -- 0169] Although Gsensor_status consists of 16-bit flags among the above-mentioned system variables, the semantics of each bit is as follows.
[0170]
[Table 1]



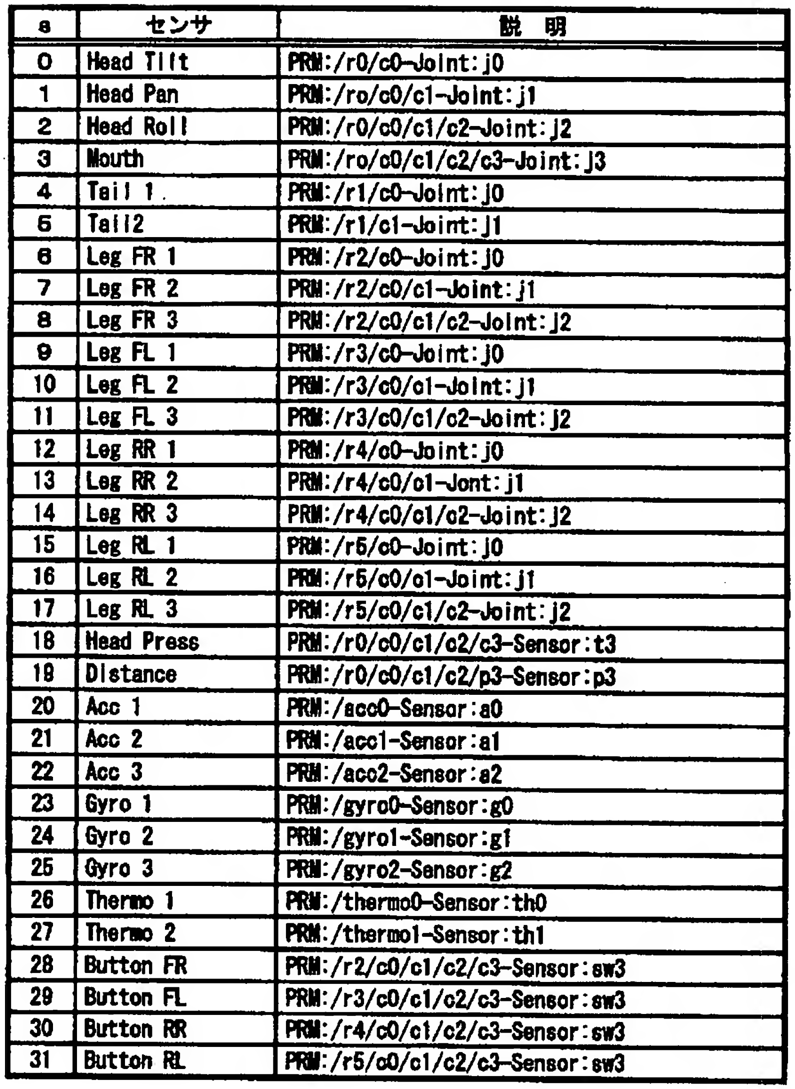
ビット	OMGsensor 状態	説　明
０ｘ０００１	FALL_DOWN_FRONT	（前方）転倒
０ｘ０００１	FALL_DOWN_RIGHT	（右側）転倒
０ｘ０００１	FALL_DOWN_LEFT	（左側）転倒
０ｘ０００１	FALL_DOWN_REAR	（後方）転倒
０ｘ０００２	HOLD_UP	抱き上げ
０ｘ０００４	HOLD_DOWN	抱き下げ
０ｘ０４００	GET_UP	転倒復帰
０ｘ０８００	JOINT_DANGER	関節挟み込み
０ｘ１０００	JOINT_GAIN_ENABLED	ゲインＯＮ
０ｘ２０００	JOINT_GAIN_DISABLED	ゲインＯＦＦ

[0171] Moreover, system variable Sound_status expresses the voice input condition to a robot 1 with the following values.
[0172]
[Table 2]



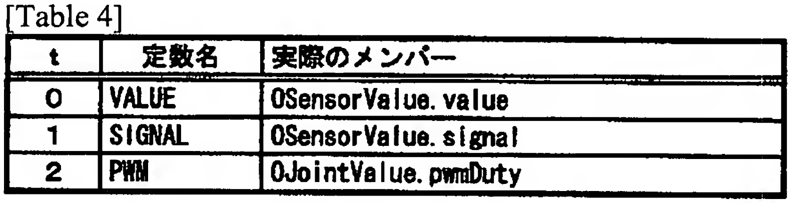
Sound_status	状　態	説　明
0	omtonePEAK	ピークのある音
1	omtoneNOPEAK	ピークのない音
2	omtoneNOPOWER	音量が小さい
3	omtoneCALC	計算中
4	omtoneOVERFLOW	オーバーフロー

[0173] Moreover, RCODE offers the service which takes the log of the output data of the sensor arranged by at least each part on a robot 1 (it stores temporarily to the predetermined field on memory). That is, while directing initiation of logging by RCODE command "LOG:START", termination of logging can be directed by RCODE command "LOG:END". Moreover, the log number of cases memorized by variable "Log_n" is expressed.
[0174] Furthermore, a log can be saved by the specified file name by RCODE command "LOG:PCFILE:<file>" at the file on a host system 100. Similarly, a log can be saved by the specified file name by RCODE command "LOG:MSFILE:<file>" at the file on a memory stick.
[0175] Moreover, it may be made to carry out a direct printout by the printer serially, without saving the output data of each sensor at memory or a file. in this case, RCODE command" -- LOG:PRINT : -- -- < -- s -- > -- : -- < -- t -- > -- : -- < -- i -- > -- " -- an output form -- it can specify . However, in s, (referring to [table 3]) and t shall show the class of value, and (referring to [table 4]) and i shall show an index (0 - Log_n -1) for a sensor number, respectively. Moreover, line feed can be directed by RCODE command "LOG:PRINT:CRLF", and file closing can be directed by RCODE command "LOG:PRINT:EOF", respectively.
[0176]
[Table 3]



s	センサ	説　明
0	Head Tilt	PRM:/r0/c0-Joint:j0
1	Head Pan	PRM:/ro/c0/c1-Joint:j1
2	Head Roll	PRM:/r0/c0/c1/c2-Joint:j2
3	Mouth	PRM:/ro/c0/c1/c2/c3-Joint:j3
4	Tail 1	PRM:/r1/c0-Joint:j0
5	Tail2	PRM:/r1/c1-Joint:j1
6	Leg FR 1	PRM:/r2/c0-Joint:j0
7	Leg FR 2	PRM:/r2/c0/c1-Joint:j1
8	Leg FR 3	PRM:/r2/c0/c1/c2-Joint:j2
9	Leg FL 1	PRM:/r3/c0-Joint:j0
10	Leg FL 2	PRM:/r3/c0/c1-Joint:j1
11	Leg FL 3	PRM:/r3/c0/c1/c2-Joint:j2
12	Leg RR 1	PRM:/r4/c0-Joint:j0
13	Leg RR 2	PRM:/r4/c0/c1-Jont:j1
14	Leg RR 3	PRM:/r4/c0/c1/c2-Joint:j2
15	Leg RL 1	PRM:/r5/c0-Joint:j0
16	Leg RL 2	PRM:/r5/c0/c1-Joint:j1
17	Leg RL 3	PRM:/r5/c0/c1/c2-Joint:j2
18	Head Press	PRM:/r0/c0/c1/c2/c3-Sensor:t3
19	Distance	PRM:/r0/c0/c1/c2/p3-Sensor:p3
20	Acc 1	PRM:/acc0-Sensor:a0
21	Acc 2	PRM:/acc1-Sensor:a1
22	Acc 3	PRM:/acc2-Sensor:a2
23	Gyro 1	PRM:/gyro0-Sensor:g0
24	Gyro 2	PRM:/gyro1-Sensor:g1
25	Gyro 3	PRM:/gyro2-Sensor:g2
26	Thermo 1	PRM:/thermo0-Sensor:th0
27	Thermo 2	PRM:/thermo1-Sensor:th1
28	Button FR	PRM:/r2/c0/c1/c2/c3-Sensor:sw3
29	Button FL	PRM:/r3/c0/c1/c2/c3-Sensor:sw3
30	Button RR	PRM:/r4/c0/c1/c2/c3-Sensor:sw3
31	Button RL	PRM:/r5/c0/c1/c2/c3-Sensor:sw3

[0177]
[Table 4]



t	定数名	実際のメンバー
0	VALUE	OSensorValue.value
1	SIGNAL	OSensorValue.signal
2	PWM	OJointValue.pwmDuty

[0178] If the authoring tool concerning this example is used and the scenario for a robot's 1 motion control is created and edited as already explained, automatic conversion will be carried out at RCODE for realizing the edited scenario. For example, the control program of the scenario for soccer games with the main routine generated on the GUI screen as shown in drawing 13 of operation is described as follows using RCODE as shown below.
[0179]
[Equation 4]
SETV:ROBOT:POWER: 1 SET:Trace:1 SET:mode:0// 0: Search mode 1: Run after and it is mode SET:head:0//. Counter SET:lost:0//for acting as the guru guru of the head by

search mode The count which missed the ball : 100//MainLoopSET:stat:Gsensor_statusAND:stat:1IF:=:stat:1:9000//Did it slip down? IF: =:mode:0:1000 It IF:=:mode:1:2000//Runs after to//search mode, and is GO:100:1000//mode:0 to the mode. Carrying out search mode SET:mode:0 MOVE:LEGS:STEP:RIGHT_TURN:0:4// right-handed rotation A ball SWITCH:head// head to look for Guru guru ****** processing CASE: Zero: MOVE: HEAD: ABS: - 15: Zero: Zero: 500CASE: One: MOVE: HEAD: ABS: - 15: - 40: Zero: 500CASE: Two: MOVE: HEAD: ABS: - 15: - 80: Zero: 500CASE: Three: MOVE: HEAD: ABS: - 45: - 40: Zero: 500CASE: Four: MOVE: HEAD: ABS: - 45: Zero: Zero: 500CASE: Five: MOVE: HEAD: ABS: - 45: 40:0:500CASE:6:MOVE:HEAD:ABS:-45:80:0:500CASE:7:MOVE:HEAD:ABS:-15:40:0:500CASE:8:MOVE:HEAD:ABS:-15:0:0:500 ADD:head:1 //head ** Increment MOD:head:9WAITIF:<:Cdt_npixel:32:100//Ball vanity ******** It runs after and is the mode. :2000//Run after. Mode SET:mode:1IF:<:Cdt_npixel:32:1000 Are they // ball vanity ******?

MOVE:HEAD:C-TRACKING: 100 //Color trace initiation IF:>:Head_tilt:-58:2300 //It is distance presumption IF:>:Head_pan:0:2210:2220 at the include angle of the head. //It will kick, if near!

: 2210 MOVE:HEAD:HOME//So that its AGO may not be kicked (^^;) as unnatural as **** -- MOVE:LEGS:KICK:LEFT_KICK:0 MOVE:LEGS:STEP:SLOW:FORWARD:1GO:2900:2220 MOVE:HEAD:HOME//One's AGO it does not kick -- as -- (^^;). as unnatural as **** -- MOVE:LEGS:KICK:RIGHT_KICK:0 MOVE:LEGS:STEP:SLOW:FORWARD:1GO:2900:2300//With the include angle of the head CSETapproaching a ball : >: Head_pan: 60: 1CSET: >: Head_pan: 45: 2CSET(s): >: Head_pan: 15: 3CSET(s): <: Head_pan: - 60: 4CSET(s): <: Head_pan: - 45: 5CSET(s): <: Head_pan: - 15: 6CSET(s): = : Zero: Zero: 0CASE: Zero: MOVE: LEGS: STEP: SLOW: FORWARD: 4CASE: One: MOVE: LEGS: STEP: SLOW: FORWARD: 4CASE: Two: MOVE: LEGS: STEP: LEFT_TURN: Zero: 4CASE: Three: MOVE: LEGS: STEP: SLOW: LEFT: 4CASE: Four: MOVE: LEGS: STEP: SLOW: LEFTFORWARD: 4CASE: Five: MOVE: LEGS:STEP:RIGHT_TURN:0:4CASE:6:MOVE:LEGS:STEP:SLOW:RIGHT:4CASE:7 :MOVE:LEGS:STEP:SLOW:RIGHTFORWARD:4:2900WAITGO:100:9000 // Fall recovery QUIT:ROBOTMOVE:ROBOT:ReactiveGUWAITGO:100[0180] Moreover, as explained referring to drawing 5 , in the authoring system concerning this example, a high level language like C can be used, for example, and a control program of operation can be created and edited in a script format. The script which realizes a function and operations sequence equivalent to the RCODE program shown with the above-mentioned [-four number] is shown below. However, the script shall be described according to the syntax of C.

[0181]

[Equation 5]

// // Soccer game////#define ROBOT_NEUTRAL described in the script format (0) // Neutral//#define ROBOT_FOUND (1) // Ball **********//#define ROBOT_LOST (2) // //#define ROBOT_FALL which is not from ball **** (99)//ROBOT fall// Pink//#define PinkBall (32) // System variable extern int Head_tilt, Head_roll;extern int

Head_pan;extern int Cdt_npixel;extern int Gsensor_status;// Whether the ball was found or it missed int bBall = 0, bBoil;// Current condition int nSoccer = 0;int nLost = 0; // A ball Number-of-steps//when missing The front declaration int of a function SoccerDog(int);void NeutralROBOT();void FindBall();void SearchBall();void RecoveryGU();void NearBall();void KickBall();// Maine void main (void) -- {-- // Soccer flag int bSoccerFlag; int dummy = 0, true = 1; bSoccerFlag = dummy = true; // A power source ON PowerOn(); // A main loop while (bSoccerFlag) -- {-- // A soccer locking dog bSoccerFlag = SoccerDog (1); } //power-source OFF PowerOff(); }// the soccer initiation int SoccerDog (int dummy) -- {-- // Divide processing in the current condition. switch(nSoccer) { case 0 : //ROBOT_NEUTRAL : // neutral condition NeutralROBOT();break;case 1://ROBOT_FOUND : // Found the ball. FindBall(); break; case 2: //ROBOT_LOST: // Missed the ball. SearchBall(); break; case 99: //ROBOT_FALL: // The Aibo fall RecoveryGU(); // Fall return processing } //continuation return 1; the processing void FindBall (void) which finds //FALSE;}// ball and approaches -- {-- int nHead;nHead = Head_tilt; // The Kalat racking RcodeColorTracking(100); // The distance of a ball at an angle of the head Measurement if(nHead > -58) {// A ball is detected. if(Cdt_npixel < 32) {// Missed the ball. nSoccer = 2; //ROBOT_LOST; } else -- {-- // A ball is approached. NearBall(); } } else -- {-- // A ball is kicked. Ki ckBall(); } }// void NearBall (void) approaching a ball -- {-- // Set up the location of a ball at an angle of a neck. if (Head_pan == 0) -- {-- RcodeStepWalk (0, 1, 4); //StepWalk(FORWARD, 4); RcodeWait(); return; } if(Head_pan> 60) { RcodeStepWalk(0, 1, 4);// StepWalk(FORWARD, 4);RcodeWait();return; } if (Head_pan > 45) -- {-- RcodeStepWalk (0, 13, 4); // StepWalk(LEFT_TURN, 4); RcodeWait();return; } if(Head_pan> 15) {RcodeStepWalk (0, 5, 4); // StepWalk (LEFT, 4); RcodeWait(); return; } if (Head_pan <-60) -- {-- RcodeStepWalk (0, 3, 4); // StepWalk(LEFTFORWARD, 4); RcodeWait(); return;} if (Head_pan <-45) -- {-- RcodeStepWalk (0, 1, 4); // StepWalk(RIGHT_TURN, 4); RcodeWait(); return;} if(Head_pan <-15) { RcodeStepWalk (0, 4, 4); //StepWalk(RIGHT, 4); RcodeWait(); return; } // Detect a ball. if -- {(Cdt_npixel < 32) -- nSoccer = 2;//ROBOT_LOST : return; }}// void KickBall (void) which kicks a ball -- {-- // Justification of the head RcodeHead (0, 0, 0, 0); // a ball -- right and left of the head -- the guide peg kicked by in which it is is decided if(Head_pan> 0) { RcodeKick(15 0); // A left leg }else -- {-- RcodeKick(14 0); // Right leg } // Since it shifts to a walk, 1 step **** RcodeStepWalk (0, 0, 1); //StepWalk(FORWARD, 1);}// the processing void SearcBall (void) which looks for a ball -- {-- // Move the head. switch (nLost) -- {-- case 0: RcodeHead(-15, 0, 0, and 500); break; case 1: RcodeHead(-15, -40, 0, and 500); break; case 2 : RcodeHead (-15, -80, 0, and 500); break;case 3 : RcodeHead (-45, -40, 0, 500); break; case 4 : RcodeHead(-45, 0, 0, and 500); break; case 5 : RcodeHead(-45, 40, 0, and 500); break; case 6 : RcodeHead(-45, 80, 0, and 500); break; case 7: RcodeHead (-15, 40, 0, and 500); break; case 8: RcodeHead(-15, 0, and 0,500);break; } // A setup of the number of steps nLost = % (nLost+1) 9; // Walk on right-hand side. RcodeStepWalk (12, 1, 4); //StepWalk(RIGHT_TURN, 4); // Color detection if -- {(Cdt_npixel < 32) -- nSoccer = 2; //ROBOT_LOST; } else{ // Initialization of nLost nLost = 0; // Processing approaching ROBOT nSoccer = 1;//ROBOT_FOUND; }}// // the processing void NeutralROBOT at the time of a neutral (void) -- {-- // fall detection if {(Gsensor_status & 1) // fall nSoccer = 99; //ROBOT_FALL; return;} // The fall has not been carried out. Ball detection if --

{(Cdt_npixel < 32) -- // Missed the ball. nSoccer = 2; //ROBOT_LOST; } else -- {-- // the ball was found nSoccer = 1; //ROBOT_FOUND; } }// the fall return void RecoveryGU (void) -- {-- // Fall return success.
RcodePlay ("ReactiveGU"); RcodeWait(); // A current condition is made neutral. nSoccer = 0; //ROBOT_NEUTRAL;} [0182] As for the authoring system concerning this example, a user can do automatic conversion of creation and the edited script at the program of a RCODE format using a text editor etc. (above-mentioned). Therefore, automatic conversion of the script shown in [the-five number] is carried out at the program code of a RCODE format as shown in [the-four number] using the authoring tool concerning this example.
[0183] For example, wireless data transmission [like bluetooth or .11B] whose host system 100 which edits a processing program of operation using RCODE and robot 1 which performs a processing program of operation are

---

[Translation done.]

* NOTICES *

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is drawing having shown the appearance configuration of the mobile robot 1 which performs the leg formula walk by the limbs with which operation is presented in this invention.
[Drawing 2] It is drawing having shown typically the block diagram of the electrical and electric equipment and control network of a mobile robot 1.
[Drawing 3] It is drawing having shown the configuration of a control section 20 in the detail further.
[Drawing 4] It is drawing having shown typically the example of a hardware configuration of a host computer 100.
[Drawing 5] It is drawing having shown typically the whole authoring system configuration concerning this example.
[Drawing 6] It is drawing having shown the configuration of a GUI screen for a user to create and edit a robot's 1 scenario of operation which the authoring tool concerning this example offers.
[Drawing 7] It is drawing having shown the pull down menu about a menu item "a file."
[Drawing 8] It is drawing having shown the pull down menu about a menu item "a group."
[Drawing 9] It is drawing having shown signs that a box [A] of operation and one

branching box [B] were arranged each on the edit window.

[Drawing 10] It is drawing having shown signs that the box [A] of operation and the branching box [B] were connected to the [START] box, the box [A] of operation, and the list, on the edit window shown in drawing 9 .

[Drawing 11] It is drawing having shown the dialog for specifying the detail of a box of operation.

[Drawing 12] It is drawing having shown the dialog for specifying the detail of a branching box.

[Drawing 13] It is drawing having shown signs that the scenario for a robot 1 to perform a soccer game was created on the GUI screen of an authoring tool.

[Drawing 14] It is drawing having shown signs that the group box in the scenario of the soccer game shown in drawing 13 [the search mode] was opened as another edit window.

[Drawing 15] It is drawing having shown signs that the group box in the scenario of the soccer game shown in drawing 13 [a trace mode] was opened as another edit window.

[Drawing 16] It is drawing having shown the configuration of the script edit window which the authoring tool concerning this example prepares.

[Drawing 17] It is drawing having shown the directed graph which specifies a robot's 1 posture held in the RCODE program execution environment by the side of a robot 1, and transition of operation.

[Description of Notations]

1 -- Mobile robot
2 -- Idiosoma unit
3 -- Head unit
4 -- Tail
6A-6D -- Leg unit
7 -- Neck joint
8 -- Tail joint
9A-9D -- Femoral region unit
10A-10D -- Leg part unit
11A-11D -- Hip joint
12A-12D -- Knee joint
15 -- CCD camera
16 -- Microphone
17 -- Loudspeaker
18 -- Touch sensor
20 -- Control section
21 -- CPU
22 -- RAM
23 -- ROM
24 -- Nonvolatile memory
25 -- Interface
26 -- Radio interface
27 -- Network Interface Card
28 -- Bus
29 -- Keyboard
40 -- I/O section

50 -- Mechanical component
51 -- Motor
52 -- Encoder
53 -- Driver

[Translation done.]

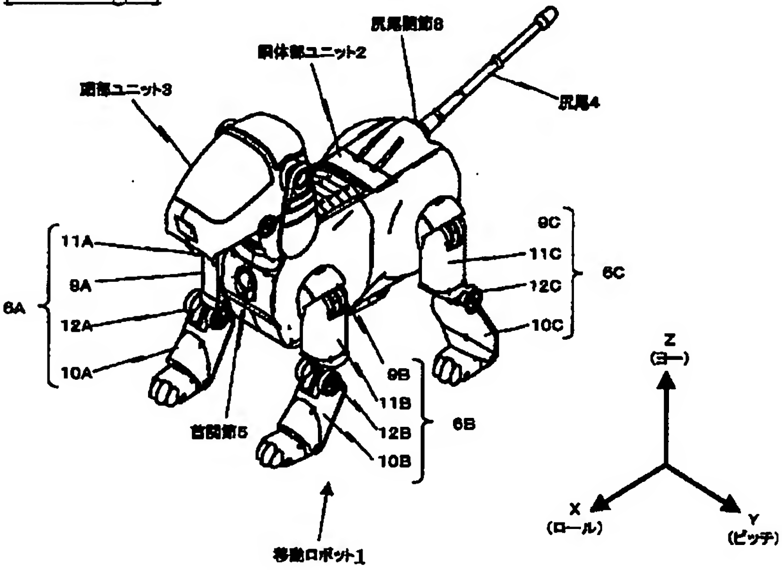
* NOTICES *

```
JPO and INPIT are not responsible for any
damages caused by the use of this translation.
```

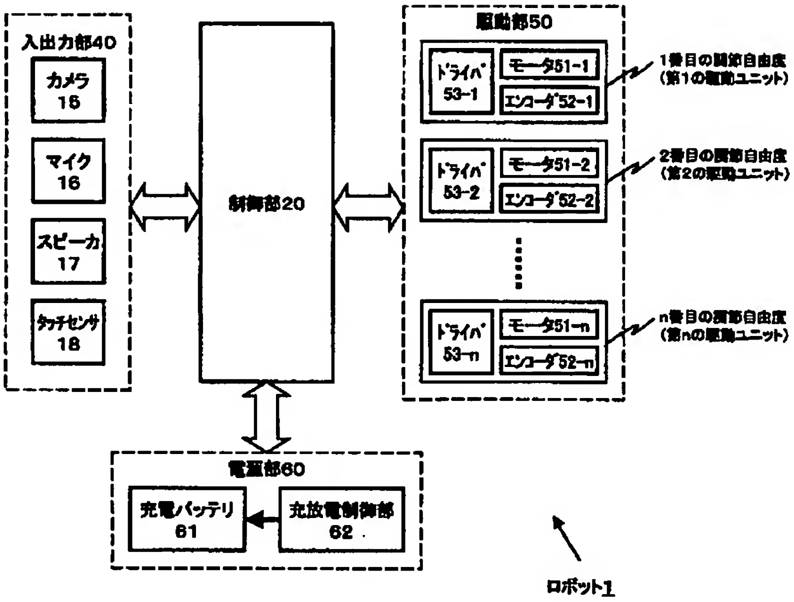
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

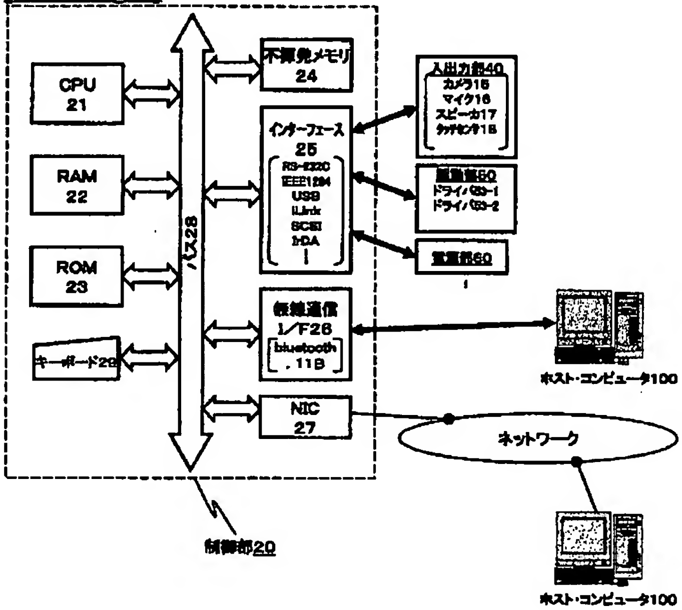
[Drawing 1]



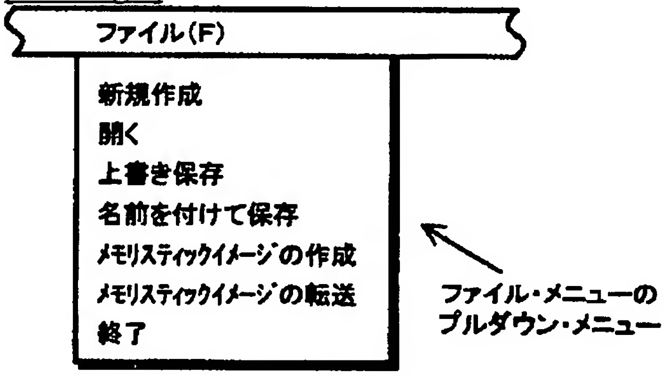
[Drawing 2]



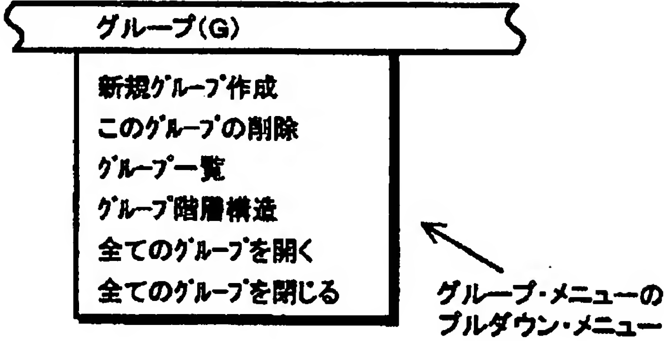
[Drawing 3]



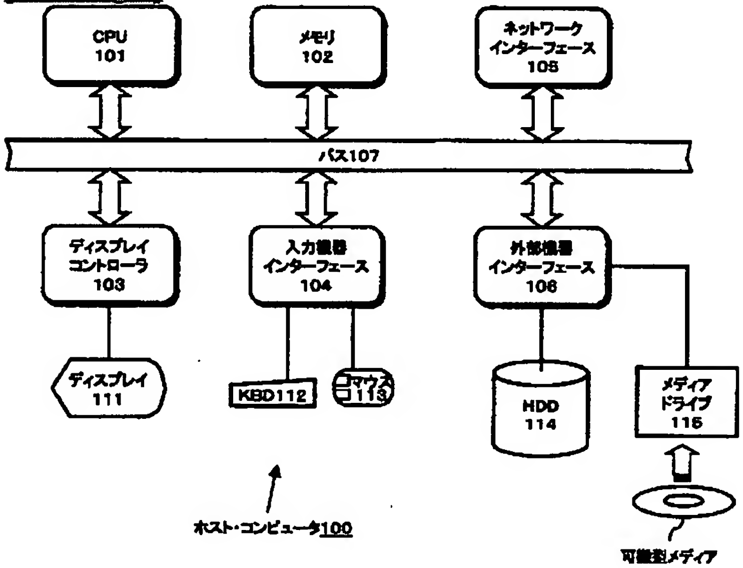
[Drawing 7]



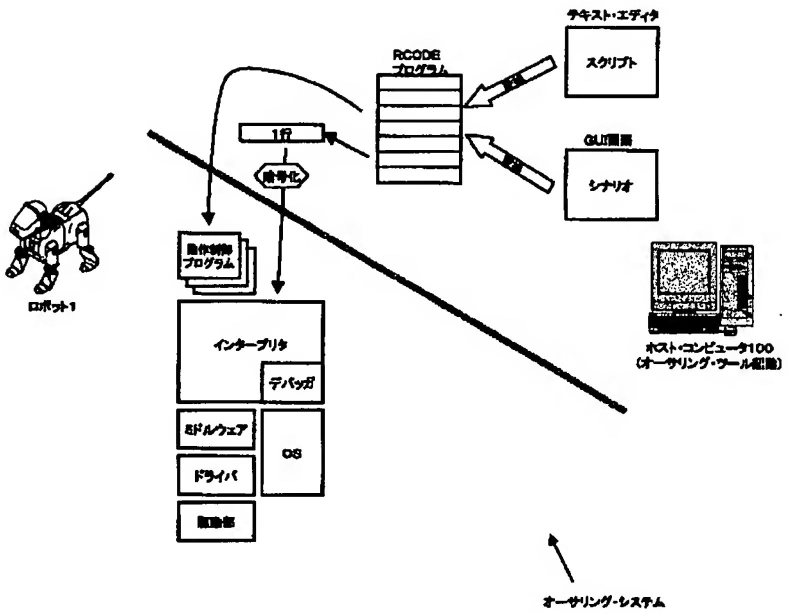
[Drawing 8]



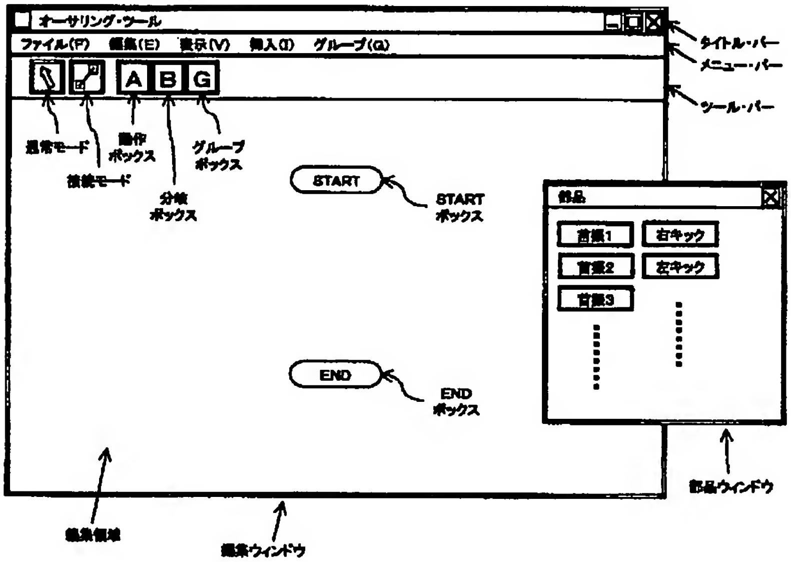
[Drawing 4]



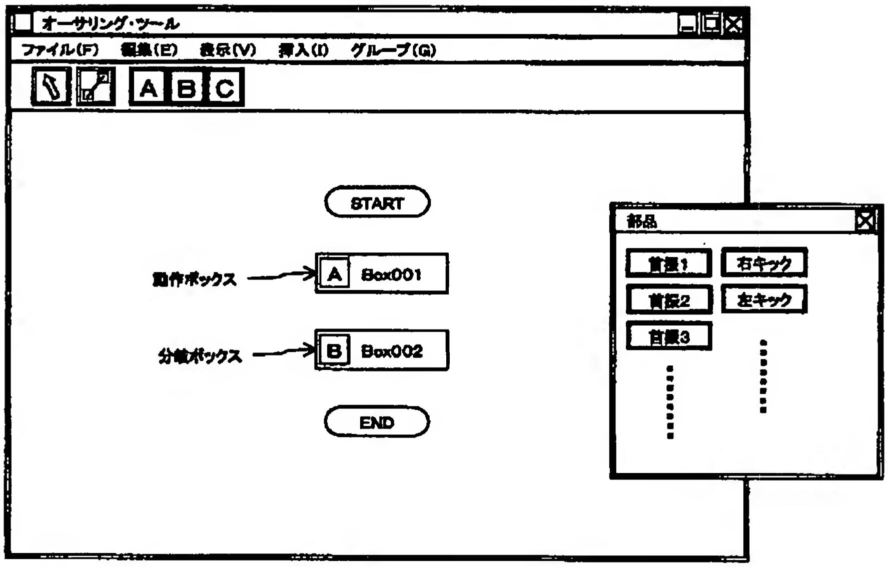
[Drawing 5]



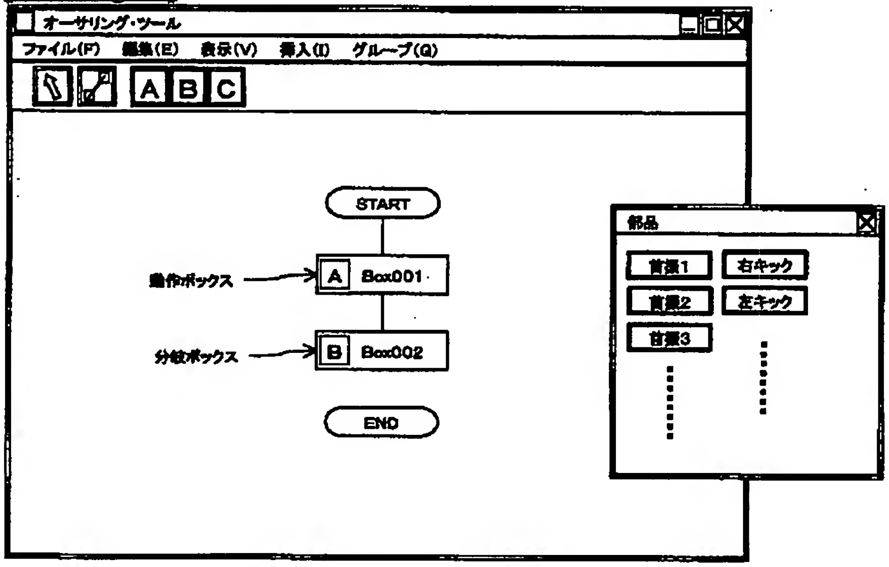
[Drawing 6]



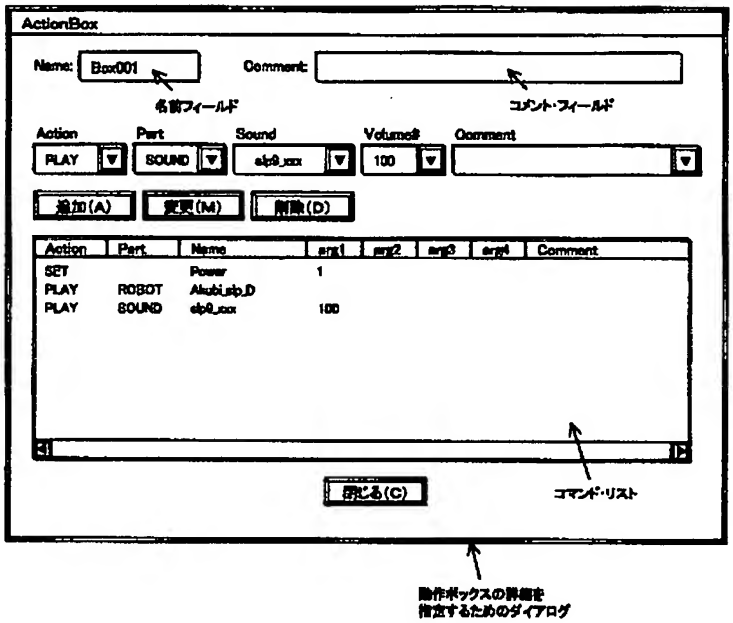
[Drawing 9]



[Drawing 10]



[Drawing 11]



ActionBox

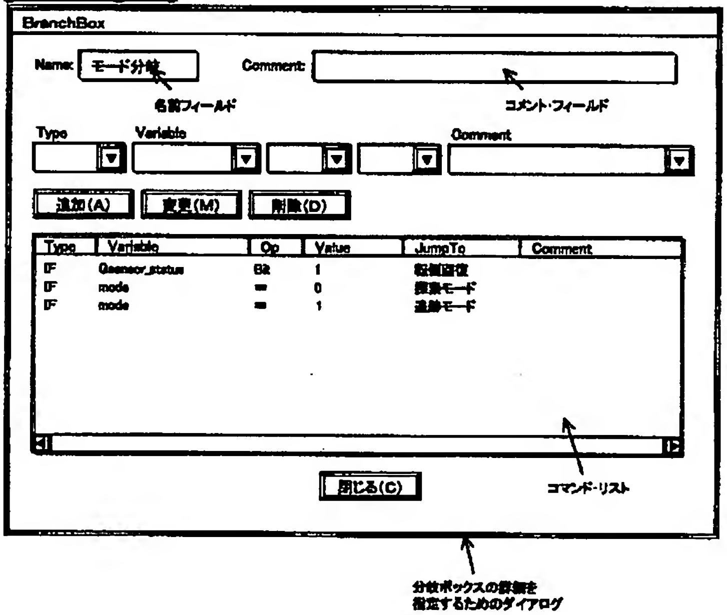
Name: Box001

Comment:

Action	Part	Sound	Volume#	Comment
PLAY	SOUND	slp9_xxx	100	

Action	Part	Name	arg1	arg2	arg3	arg4	Comment
SET		Power	1				
PLAY	ROBOT	Akubi_slp_D					
PLAY	SOUND	slp9_xxx	100				

[Drawing 12]



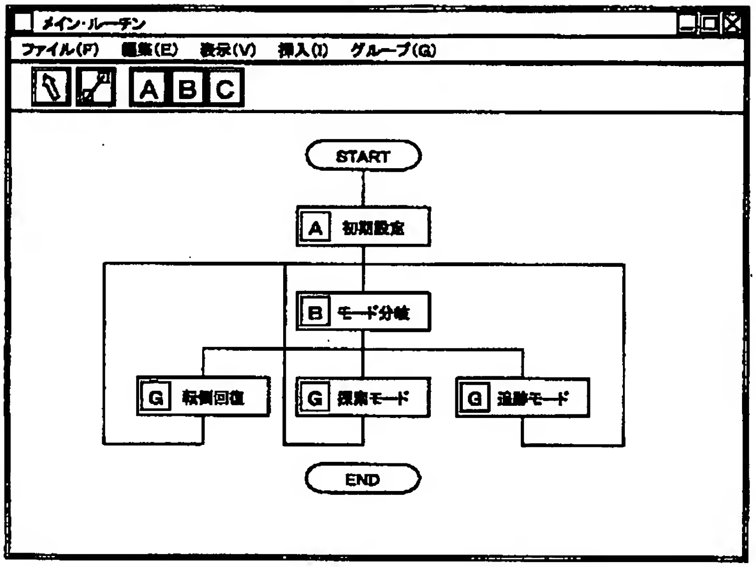
BranchBox

Name: モード分岐

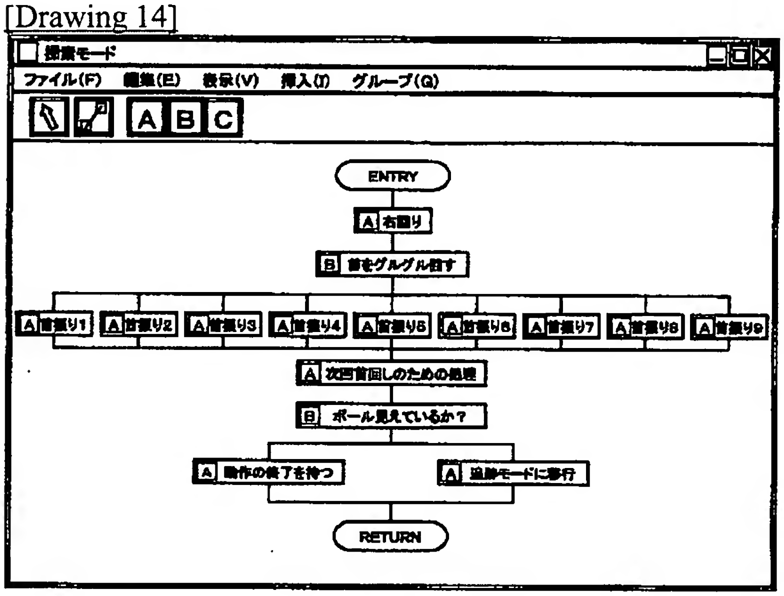
Comment:

Type	Variable	Op	Value	JumpTo	Comment
IF	Gsensor_status	Bit	1	転倒回復	
IF	mode	==	0	探索モード	
IF	mode	==	1	追跡モード	

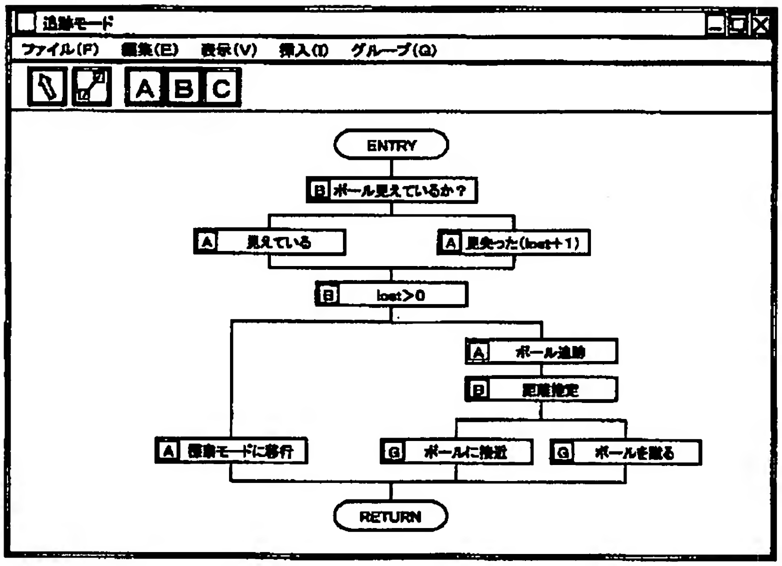
[Drawing 13]



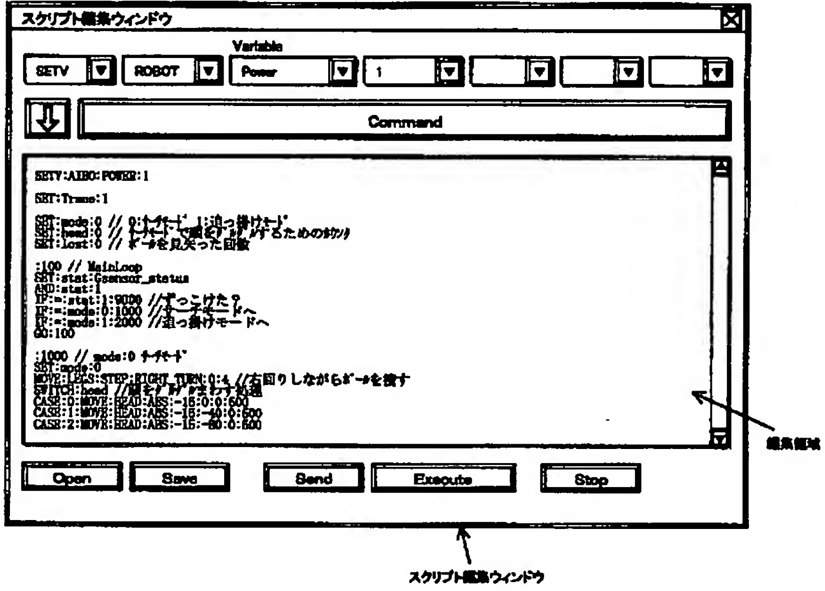
[Drawing 14]



[Drawing 15]

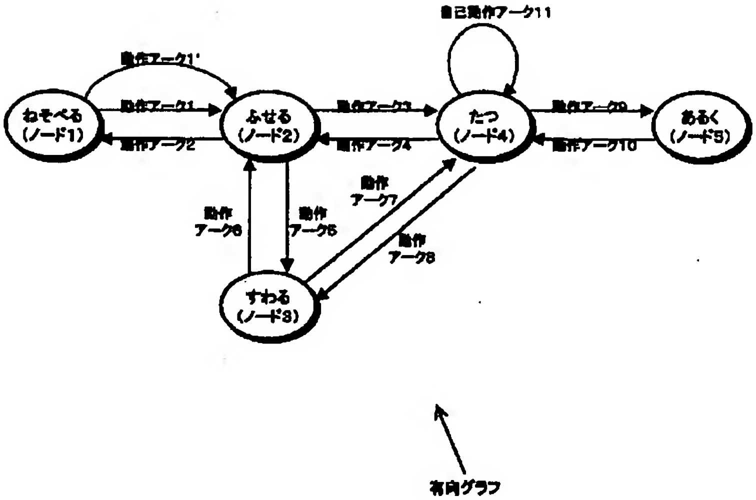


[Drawing 16]



```
SETV:AIBO:POWER:1

SET:Trace:1

SET:mode:0 // 0:ｻｰﾁﾓｰﾄﾞ 1:追っ掛けﾓｰﾄﾞ
SET:head:0 // ｻｰﾁﾓｰﾄﾞで頭をﾌﾞﾙﾌﾞﾙするためのｶｳﾝﾀ
SET:lost:0 // ﾎﾞｰﾙを見失った回数

:100 // MainLoop
SET:stat:Gsensor_status
AND:stat:1
IF:=:stat:1:9000 //ずっこけた？
IF:=:mode:0:1000 //サーチモードへ
IF:=:mode:1:2000 //追っ掛けモードへ
GO:100

:1000 // mode:0 ｻｰﾁﾓｰﾄﾞ
SET:mode:0
MOVE:LEGS:STEP:RIGHT_TURN:0:4 //右回りしながらﾎﾞｰﾙを探す
SWITCH:head //頭をﾌﾞﾙﾌﾞﾙまわす処理
CASE:0:MOVE:HEAD:ABS:-15:0:0:600
CASE:1:MOVE:HEAD:ABS:-15:-40:0:600
CASE:2:MOVE:HEAD:ABS:-15:-80:0:600
```

[Drawing 17]



[Translation done.]